On the Security of Protocols with Logarithmic Communication Complexity

Michael Backes and Dominique Unruh

Saarland University, Saarbrücken, Germany
{backes,unruh}@cs.uni-sb.de

Abstract. We investigate the security of protocols with logarithmic communication complexity. We show that for the security definitions with environment, i.e., Reactive Simulatability and Universal Composability, computational security of logarithmic protocols implies statistical security. The same holds for advantage-based security definitions as commonly used for individual primitives. While this matches the folklore that logarithmic protocols cannot be computationally secure unless they are already statistically secure, we show that under realistic complexity assumptions, this folklore does surprisingly *not* hold for the stand-alone model without auxiliary input, i.e., there are logarithmic protocols that are statistically insecure but computationally secure in this model. The proof is conducted by showing how to transform an instance of an NP-complete problem into a protocol with two properties: There exists an adversary such that the protocol is statistically insecure in the stand-alone model, and given such an adversary we can find a witness for the problem instance, hence yielding a computationally secure protocol assuming the hardness of finding a witness. The proof relies on a novel technique that establishes a link between cryptographic definitions and foundations of computational geometry, which we consider of independent interest.

Table of Contents

1 Introduction

In this work, we investigate the security of cryptographic protocols with logarithmic communication complexity (logarithmic protocols for short). The central question we are aiming to solve is the following: Are there logarithmic protocols that are computationally secure but not statistically (information-theoretically) secure, i.e., can we base the security of logarithmic protocols on suitable complexity assumptions? At first glance, the answer seems obviously negative and constitutes a folklore in cryptography: If a protocol is not statistically secure anyway, and if all messages have logarithmic length, the protocol can be efficiently attacked by brute-force and hence cannot be computationally secure. We investigate whether this folklore indeed withstands a formal investigation. (Anticipating the answer: No, it does not in general.)

We consider the question in three different security models: security with environment, stand-alone security and advantage-based security. Security with environment is a family of very stringent security definitions, out of which the Reactive Simulatability framework and the Universal Composability framework constitute the most prominent members. Because of strong compositionality results, security with environment has rapidly gained momentum in the last years. Stand-alone security on the other hand does not entail such strong compositionality guarantees, but it allows to derive suitable security guarantees for many cryptographic protocols for which security with environment is too strong a notion. Stand-alone security thus still constitutes one of the standard security notions in cryptography. Both security with environment and stand-alone security define security by comparing a protocol with some ideal specification. This intuitively guarantees that all properties enjoyed by the ideal specification are also fulfilled by the real protocol. In contrast, advantage-based security notions define a particular concrete property the protocol must satisfy. More precisely, one specifies a game and a well-defined goal, and then requires that every adversary only attains that goal with a sufficiently small probability (the so-called advantage). Stand-alone security is often seen as—and in fact was designed with the intuition of being—the union of all security properties fulfilled by the ideal specification. In other words, one expects a protocol to be stand-alone secure if for any advantage-based security notion that is fulfilled by the ideal specification, the real protocol also fulfils this property.

In the case of security with environment and of advantage-based security, we show that the folklore statement indeed holds true: For these notions, computational security implies statistical security. In the case of security with environment we prove this by showing that adversaries that randomly choose their communication are complete for logarithmic protocols, i.e., if there is some (possibly unbounded) adversary breaking the protocol, then the adversary using randomly selected messages also breaks the protocol. In the case of advantage-based security we analyse the protocol in a game-theoretic setting and show that an optimal strategy can efficiently be computed.

Most interestingly, and more surprisingly, we show that in the case of stand-alone security, the folklore statement *does not hold* in the case without auxiliary input. We give a reduction that allows to convert an instance of the NP-complete set cover problem into a protocol with the following property: If the set cover instance has a witness, then there exists a successful adversary and the protocol is not statistically secure, and given such an adversary, we can extract a witness for the set cover instance. The consequence is that if finding witnesses for set cover is hard (more exactly, if $\mathrm{NP} \not\subseteq \mathrm{BPTIME}(n^{O(\log n)})$ in our specific case), finding a successful efficient

adversary is hard, too. In order to show that it is not only hard to find an adversary, but even that no efficient adversary exists, we additionally assume that efficiently computable sequences of hard instances of some NP-problem exist. We then construct a protocol that uses one of these instances for each security parameter. A successful efficient adversary would consequently be able to solve infinitely many of the hard instances, yielding a contradiction. Hence the resulting protocol is computationally secure but not statistically secure. This argumentation also holds for a *uniform* auxiliary input. However, in the case of *nonuniform* auxiliary input (in the sense of [Gol93]) the argument fails since we can encode the witness into the auxiliary input.

This separation has several interesting implications. First, it shows that the proof idea of breaking any logarithmic protocol with brute force does not work in general and that there are cryptographic problems that are more than exponentially hard in the length of the communication. Second, since we showed that for advantage-based security notions computational implies statistical security, it follows that stand-alone security is more that just the union of all advantage-based security properties fulfilled by the ideal specification. This stands in contrast to the folklore point of view mentioned above, and it can even be seen as evidence that the intuition underlying the stand-alone model has not been fully met. Arguably the most interesting implication is the third one: Another folklore theorem states that if $\mathrm{P} = \mathrm{NP}$ (or $\mathrm{BPP} = \mathrm{MA}$ to be more exact), cryptography becomes generally insecure in the sense that every statistically insecure protocol is also computationally insecure. However, the intuition underlying this statement is similar to the intuition of using a brute-force attack to break any logarithmic protocol. As we have shown the latter intuition to be unsound, it may be that a similar approach might also show the first one to be incorrect, i.e., it might be the case that even if $\mathrm{P} = \mathrm{NP}$ and $\mathrm{BPP} = \mathrm{MA}$, computationally secure protocols exist that are not statistically secure.

Related Work. The paper that comes closest to our work is [Unr06]. There it was shown that for security with environment and polynomial-time protocols, statistical security and security with respect to exponential-time adversaries coincide. This is analogous to our result for the setting of security with environment, only one level higher in the complexity hierarchy. Note however that directly applying their technique to the setting of logarithmic protocols yields a weaker result than the one we achieve when dealing with security with environment: For protocols that have logarithmic communication complexity *and run in logarithmic time*, computational security with environment implies statistical security with environment. However, the results in [Unr06] still served as the inspiration for analysing the security of logarithmic protocols.

Additionally relevant for our work are the various security models for dealing with cryptographic primitives. The idea of using a simple ideal system as a specification for a cryptographic system was first sketched for secure multi-party function evaluation, i.e., for the computation of one output tuple from one tuple of secret inputs from each participant in [Yao82] and defined (with different degrees of generality and rigorosity) in [GL90, Bea91, MR91, Can95, Can00, Gol04]. These models are currently jointly denoted the *stand-alone model* of cryptography. Extensions of this idea to specific reactive problems were first given in [GM95, BCK98, CG99] but without a detailed or general definition. In a similar way, construction of generic solutions for large classes of reactive problems were proposed [GMW87, Gol98, HM00], but usually yielding inefficient solutions and assuming that all parties take part in all subprotocols. The currently prevalent frameworks for dealing with reactive protocols are the Reactive Simulatability (RSIM)

framework [PW01, BPW04] and the Universal Composability (UC) framework [Can01, Can05], which both pursue the idea of augmenting the stand-alone model with an environment that essentially ensures security in arbitrary surrounding contexts in which the protocol under consideration is executed. This *security with enviroment* can be shown to entail strong compositionality guarantees and has proven successful in analyzing various cryptographic primitives and protocols. *Advantage-based definitions* of cryptographic primitives have been playing a key role from the very start in essentially all cryptographic definitions, e.g., semantic security [GM84], CMA-security of signatures [GMR88], and many more.

Outline. In Section 2, we present the notation and security definitions used in the subsequent sections. In Section 3 we give an intermediate result: If random variables of logarithmic length are computationally indistinguishable, they are also statistically indistinguishable. In Section 4 we show that for logarithmic protocols, computational security with environment implies statistical security with environment. In Section 5 we show that for stand-alone security, this does not hold in general. We construct logarithmic protocols that are computationally stand-alone secure without auxiliary input but not statistically stand-alone secure. In the presence of auxiliary input, we show that if a logarithmic protocol is computationally stand-alone secure, it is also statistically stand-alone secure. In Section 6 we show that for advantage-based security notions, computational security implies statistical security.

2 Notation and Security Models

Notation. The *real numbers* are denoted $\mathbb{R}$, the *natural numbers* by $\mathbb{N} = \{1, 2, \dots\}$. The *statistical distance* between X and Y we denote $\Delta(X; Y)$. Two families of random variables $\{X_z\}_{z\in Z}$ and $\{Y_z\}_{z\in Z}$ are *statistically indistinguishable* if $\Delta(X_z; Y_z)$ is negligible in $|z|$. The families $\{X_z\}_{z\in Z}$ and $\{Y_z\}_{z\in Z}$ are *computationally indistinguishable* if for any probabilistic polynomial-time algorithm D the difference $|\Pr[D(z, X_z) = 1] - \Pr[D(z, Y_z) = 1]|$ is negligible in $|z|$. A family $\{X_z\}_{z\in Z}$ is *efficiently constructible* if there is a probabilistic polynomial-time algorithm S such that $S(z)$ has distribution X_z. If $Z = \mathbb{N}$, we interpret $z \in \mathbb{N}$ as its unary encoding 1^z.

If A and B are interactive Turing machines (ITMs), we write $\langle A, B\rangle$ for the output *of* B in an execution of A and B. We write $\langle\langle A, B\rangle\rangle$ for the pair consisting of the outputs *of* A *and* B. If A and B take some input x and y, we write $\langle A(x), B(y)\rangle$ and $\langle\langle A(x), B(y)\rangle\rangle$.

For two vectors $x, y \in \mathbb{R}^n$, we write $\langle x, y\rangle := \sum_i x_i y_i$ for their *inner product*. The l_1*-norm* of x we write $\|x\|_1 := \sum_i |x_i|$. The l_1*-distance* is written $d_1(x, y) := \|x - y\|_1$. For a matrix $S := (s_{ij}) \in \mathbb{R}^{m\times n}$, let $s_{i\cdot}$ denote its ith row. Given two sets $X, Y \subseteq \mathbb{R}^n$ and a scalar $\alpha \in \mathbb{R}$, we write $X + Y := \{x + y : x \in X, y \in Y\}$ and $\alpha X := \{\alpha x : x \in X\}$. A subset $X \subseteq \mathbb{R}^n$ is a *halfspace* if it has the form $X = \{x : \langle c, x\rangle \leq b\}$, and X is called a *polytope* if it is bounded and the intersection of finitely many halfspaces.

Security models. An important class of security models are the *security models with environment*, its best-known representatives being the Reactive Simulatability (RSIM) framework [BPW04] and the Universal Composability (UC) framework [Can05]. In the Reactive Simulatability framework, we consider an execution of a protocol π together with an adversary A and

an honest user H (also known as the environment). The sequence of all internal states of H and messages sent and received by H is called its view and written $view_{\pi,A,H,k}(H)$. Here $k \in \mathbb{N}$ is the security parameter available to all machines. For a detailed definition we refer to [BPW04]. In the Reactive Simulatability framework, security is then defined as follows:

Definition 1 (Reactive Simulatability (sketch)). *A protocol π is as secure as a protocol ρ with respect to* computational universal reactive simulatability *if for every polynomial-time machine A (the adversary) there is a polynomial-time machine S (the simulator) such that for every polynomial-time machine H (the honest user) $\{view_{\pi,A,H,k}(H)\}_{k\in\mathbb{N}}$ and $\{view_{\rho,S,H,k}(H)\}_{k\in\mathbb{N}}$ are computationally indistinguishable in k.*

We speak of statistical *universal reactive simulatability if in the above definitions A, H and S are unbounded and statistical indistinguishability is used instead of computational indistinguishability.*

Other variants of security models with environment exist, e.g. *general* reactive simulatability where the simulator may depend on the honest user [BPW04], and UC security, which is similar to Definition 1 but formulated in the UC framework [Can05].

Another very common security definition is *stand-alone security*. It is weaker than the security models with environment, and many useful protocols are only stand-alone secure. Since there are many variants of stand-alone security (e.g., [Can95, Gol04]), we work with the following generalised definition.

Definition 2 (Stand-Alone Security). *Let π and ρ be ITMs. We say that π* is as secure as *ρ with respect to* computational stand-alone security with auxiliary input, *if for every polynomial-time ITM A (the adversary) there is a polynomial-time ITM S (the simulator) such that for sequences x and z of strings of polynomial length, the families of distributions $\{\langle\!\langle A(1^k, z_k), \pi(1^k, x_k)\rangle\!\rangle\}_{k,z_k,x_k}$ and $\{\langle\!\langle S(1^k, z_k), \rho(1^k, x_k)\rangle\!\rangle\}_{k,z_k,x_k}$ are computationally indistinguishable in k.*

We speak of statistical stand-alone security with auxiliary input *if the above holds with unbounded A and S and statistical indistinguishability.*

We speak of computational/statistical stand-alone security without auxiliary input *if A and S do not get the additional input z_k (i.e. the distributions $\langle\!\langle A(1^k), \pi(1^k, x_k)\rangle\!\rangle$ and $\langle\!\langle S(1^k), \rho(1^k, x_k)\rangle\!\rangle$ are compared).*

Depending on the variant of stand-alone security we consider, the protocols π and ρ do not only incorporate the actual behaviour of all uncorrupted parties, but also mechanisms for delivering messages, corrupting parties and—of specific importance for the ideal model—passing inputs to the corrupted parties. In many definitions, the ideal protocol ρ is not allowed to be an arbitrary protocol, but only a probabilistic function. This can be realised by requiring ρ to receive only one message (corresponding to the input from the simulator) and to send only one message (to pass the output of the corrupted parties to the simulator). Our construction in Section 5 is of that form.

Finally, one is often not interested in protocols that are indistinguishable from some ideal protocol, but in protocols where the adversary is unable to achieve a specific goal with more than a certain probability (the advantage of the adversary). These *advantage-based* definitions can be capture by the following definition.

Definition 3 (Advantage-Based Security). *Let B be an ITM and γ a function. We say that B is* γ-secure with respect to computational advantage-based security with auxiliary input *if for every polynomial-time ITM A and for all sequences x and z of strings of polynomial length, there is a negligible function μ such that* $\Pr[\langle A(1^k, z_k), B(1^k, x_k)\rangle = 1] \leq \gamma(k) + \mu(k)$ *for all* $k \in \mathbb{N}$.

We speak of statistical advantage-based security if the above holds with unbounded A.

We speak of advantage-based security without auxiliary input if A does not get the additional input z_k (i.e., the distribution $\langle A(1^k), B(1^k, x_k)\rangle$ is considered).

In this definition, the ITM B takes the role of both the protocol under consideration and the game defining the desired security property. In the definition of, e.g., IND-CPA security, B would expect two plaintexts from A, encrypt one of them, and then output if the adversary guesses correctly which plaintext was encrypted.

3 Indistinguishability of Logarithmic Random Variables

Before analysing more complex security notions, we start by investigating the indistinguishability of random variables. For random variables of logarithmic length, statistical and computational indistinguishability coincide. This fact will be useful in the equivalence proofs for the more complex security notions.

Theorem 4 (Indistinguishability of Logarithmic Random Variables). *Let $Z \subseteq \{0,1\}^*$. Let $X = \{X_z\}_{z \in Z}$ and $Y = \{Y_z\}_{z \in Z}$ be efficiently constructible families of random variables of logarithmic length.*

If X and Y are computationally indistinguishable, then they are statistically indistinguishable.

Proof (sketch). If X and Y are statistically distinguishable, there is a polynomial p such that $\Delta(X_z, Y_z) \geq \frac{1}{p}$ for infinitely many lengths $|z|$. Since X_z and Y_z have a range of polynomial size we can approximate the distribution of X_z and Y_z using a polynomial number of samples with an expected error of $\frac{1}{q}$ where q is an arbitrary polynomial. Given an explicit description of the true distributions of X_z and Y_z, we can efficiently derive an optimal distinguisher: Upon input x, determine whether x is more likely when drawing from X_z or from Y_z. If we use the approximated distributions instead, the resulting efficient distinguisher D is not optimal anymore, but for sufficiently large q, the error introduced by the approximation is at most $\frac{1}{2p}$, so $|\Pr[D(X_z) = 1] - \Pr[D(Y_z) = 1]| \geq \Delta(X_z, Y_z) - \frac{1}{2p} \geq \frac{1}{2p}$ infinitely often. Thus X and Y are computationally distinguishable. □

4 Security with Environment

We show that for the security notions with environment (i.e., RSIM and UC) computational security implies statistical security in the case of logarithmic protocols. These notions contain two adversarial entities—the environment and the adversary. It is well-known that the latter can be assumed to be a fixed machine that just forwards messages between environment and protocol (the so-called dummy-adversary). For the environment, no such reduction is known in

general. However, in the case of logarithmic communication complexity, the set of all possible communication traces has polynomial size, so the probability of randomly guessing a given communication trace is noticeable. Then, if a (possibly unbounded) environment E succeeds in distinguishing the real and the ideal protocol, an environment $\tilde{E}$ that simply guesses all messages that E sends can be shown to be a successful distinguisher, too. This is captured in the following lemma.[1]

Lemma 5. *Let X and Y be oracle Turing machines. Let A be an oracle. Assume both X and Y call their oracle at most r times, and that the total length of the answers given by A is at most l. Assume further that all oracle queries and oracle answers can be extracted from the output of X and Y. Let $\tilde{A}$ be the oracle that first uniformly choose an r-tupel $(o_1, \ldots, o_r)$ of strings such that the total length $\sum o_i$ is at most l, and then upon its i-th activation responds with o_i. Then $\Delta(X^{\tilde{A}}; Y^{\tilde{A}}) \geq 2^{-O(l+r)}\Delta(X^A; Y^A)$.*

This lemma is proven by induction over the number of rounds.

The construction in this lemma represents the essentials of the definitions of Reactive Simulatability and UC (and probably other flavours of security with environment). The oracle A (or $\tilde{A}$) represents the environment, while X and Y represent the real and the ideal protocol execution. More exactly, the machine X contains the complete real model, including adversary, real protocol and the underlying network model, while all messages sent to the environment are realised as oracle calls to A. Similarly, the machine Y contains the simulator, the ideal protocol and the underlying network model. In this light, Lemma 5 states that (independent of adversary and simulator), we can replace any environment by an environment that randomly chooses its messages, and which hence runs in probabilistic polynomial time. Additionally exploiting that the view of the environment has logarithmic length, and hence that computational and statistical indistinguishability of the views of the environment coincide by Theorem 4, we obtain the following theorem:

Theorem 6 (Computational Implies Statistical Simulatability/UC). *Let π and ρ be polynomial-time protocols with logarithmic communication complexity. Assume that π is as secure as ρ with respect to computational universal Reactive Simulatability. Then π is as secure as ρ with respect to statistical universal reactive simulatability. The same holds for general reactive simulatability and for UC.*

5 Stand-Alone Security

Surprisingly, the results of the preceding section do not apply to the stand-alone model (without auxiliary input): Under realistic complexity assumptions, there are logarithmic protocols that are statistically insecure, but computationally secure. (In this section, security always means stand-alone security without auxiliary input.) The random adversary we used in the previous section does not work in this case as illustrated by the following example: Consider the insecure cointoss protocol where Bob randomly chooses the outcome and sends it to Alice. An adversary

[1] However, in the actual proof the factor by which the statistical distance is reduced is not the probability of guessing a given communication, but instead 3^{-r} times that probability, where r is the number of rounds. It would be interesting to know whether this is an artifact of our proof, or whether this factor is indeed necessary.

that *randomly* chooses its messages would—in the case of a corrupted Bob—choose a *random* outcome, which corresponds to Bob's honest behaviour.

To prove the separation, we give a construction that transforms a yes-instance of the set cover problem into a protocol with two properties: There exists an adversary such that the protocol is statistically insecure, and given such an adversary we can find a witness for the set cover instance. Consequently, such a protocol is computationally secure unless such witnesses can be found in probabilistic polynomial time.

Definition 7 (Set Cover). *Let $n \in \mathbb{N}$, $s_{ij} \in \{0,1\}$ with $i = 1,\ldots,m$ and $j = 1,\ldots,n$ and $d \leq n$. Let $s_{i\cdot} \neq 0$ for all $i = 1,\ldots,m$. Then (n, m, S, d) is an instance of set cover (with $S := ((s_{ij}))$). Let $S_j := \{i : s_{ij} = 1\}$. Then (n, m, S, d) is a yes-instance of set cover if there is a set $C \subseteq \{1,\ldots,n\}$ such that $\#C \leq d$ and $\bigcup_{j\in C} S_j = \{1,\ldots,m\}$.*

Set cover is well-known to be NP-complete. To describe our construction in more detail, we first define the class of good adversaries, namely those that perform an attack that cannot be simulated.

Definition 8 (Good Adversaries). *Let π and ρ and A be ITMs, and let $\varepsilon > 0$. We call A ε-*good for (π, ρ) *if for all ITMs S the statistical distance between $\langle\langle A, \pi\rangle\rangle$ and $\langle\langle S, \rho\rangle\rangle$ is bounded from above by ε. We call A* good for (π, ρ) *if A is ε-good for some $\varepsilon > 0$.* Strongly good *adversaries are defined analogously, with $\langle A, \pi\rangle$ and $\langle S, \rho\rangle$ instead of $\langle\langle A, \pi\rangle\rangle$ and $\langle\langle S, \rho\rangle\rangle$.*

Obviously, a protocol π is statistically as secure as a protocol ρ if and only if there is a negligible function ε such that $A(1^k)$ is $\varepsilon(k)$-good for $(\pi(1^k), \rho(1^k))$. The stricter notion of strongly good adversaries represents adversaries for which already the protocol output (without the adversary's/simulator's output) is distinguishable in the real and the ideal model. So intuitively, a strongly good adversary breaks the correctness and not only the secrecy of the protocol. However, the protocols we are going to construct will not keep any secrets from the adversary/simulator, so good and strongly good adversaries coincide in this case. Our goal at this point is to transform a given set cover instance into a protocol pair such that good adversaries correspond to witnesses. (At this point, we are interested in protocols that are not parametrised by the security parameter. Later, a sequence of set cover instances will be used to construct a parametrised protocol.)

For our transformation, we interpret the property of being a good adversary geometrically. With any protocol π we associate the set of all probability distributions of π's output when run with different adversaries. We consider these distributions as points in an Euclidean space as follows: for an ITM T whose output lies in the set $\{1,\ldots,t\}$, we consider $p := \langle A, T\rangle$ as a vector in $\mathbb{R}^t$ by setting $p_i := \Pr[\langle A, T\rangle = i]$. This gives rise to the following definition:

Definition 9 (Adversary-Polytope). *The* adversary-polytope $\mathbf{A}_T \subseteq \mathbb{R}^t$ *of the ITM T is defined as $\mathbf{A}_T := \{\langle A, T\rangle : A$ is an ITM$\}$.*

We can now reformulate strongly good adversaries geometrically: An adversary A is strongly good for (π, ρ) if $\langle A, \pi\rangle \notin \mathbf{A}_\rho$, and it is strongly ε-good if $d_1(\langle A, \pi\rangle, \mathbf{A}_\rho) \geq 2\varepsilon$. So the problem of finding strongly ε-good adversaries corresponds to the following geometric problem: Given two polytopes A and X (the adversary-polytopes of the real and the ideal protocol π and ρ), find a point in A that is at least ε away from X (w.r.t. the l_1-norm). In particular, if X is the set of all points p with $\|p\|_1 \leq l$ (a higher dimensional octahedron, the so-called cross-polytope), an

adversary A is strongly ε-good if and only if $\|\langle A, \pi\rangle\|_1 \geq l + \varepsilon$. So in this case, the question whether strongly ε-good adversaries exist can be reduced to the problem of estimating the size of $\mathbf{A}_\pi$ (w.r.t. the l_1-norm).[2] However, estimating the size of a polytope is hard in general: Let a set cover instance (n, m, S, d) be given. Let $P_* \subseteq \mathbb{R}^n$ be the polytope defined by the following inequalities: $x \in P_*$ if $0 \leq x_j \leq 1$, $\sum x_j \leq d$, and for all $i = 1, \ldots, m$: $\sum x_j s_{ij} \geq 1$. If we associate a point $v \in \{0,1\}^n$ with a set $C := \{i : v_i = 1\}$, it is easy to see that such a point v is in P_* if and only if C is a witness for the set-cover instance. Since $v \in [0,1]^n$ is in $\{0,1\}^n$ if and only if $d_1(v, u) = \frac{n}{2}$ where $u := (\frac{1}{2}, \ldots, \frac{1}{2})^T$, and $d_1(v, u) \leq \frac{n}{2}$ for all $v \in [0,1]^n$, if follows that $\|P_* - u\|_1 \geq \frac{n}{2}$ if and only if (n, m, S, d) is a yes-instance, and any point $v \in P_*$ with $d_1(v, u) \geq \frac{n}{2}$ gives us a witness for that set cover instance. Moreover, it turns out that approximating $\|P_* - u\|_1$ up to an additive constant is already sufficient. Unfortunately we cannot construct protocols that have P_* as their adversary-polytope (for some no-instances (n, m, S, d), P_* is empty, which cannot happen for adversary-polytopes). Fortunately, requiring the equations defining P_* to hold only approximately still allows to reduce the set cover instance to it, and the resulting polytope can be constructed as an adversary-polytope as we will see below.

Definition 10 (Set Cover Polytope). *Let $\varepsilon \in (0,1)$. Let P be a polytope. We call P an* ε-set cover polytope for (n, m, S, d) *if the following holds:*

- $P \subseteq [0,1]^n$.
- *Let $v \in \{0,1\}^n$. If $\|v\|_1 \leq d$ and $\langle s_{i\cdot}, v\rangle \geq 1$ for all $i = 1, \ldots, m$, then $v \in P$.*
- *Let $v \in [0,1]^n$. If $\|v\|_1 > d + 1 - \varepsilon$ or $\langle s_{i\cdot}, v\rangle < \varepsilon$ for some $i \in \{1, \ldots, m\}$, then $v \notin P$.*

Obviously, P as constructed above is an ε-set cover polytope for any $\varepsilon \in (0,1)$.

Lemma 11 (Reducing Set Cover to Polytope 1-Norm). *Let P be an ε-set cover polytope and let $P' := P - \frac{1}{2}u$. Then there is a $\delta \in \Omega(\varepsilon/\mathrm{poly}(n))$ such that*

(i) If (n, m, S, d) is a yes-instance, then $\|P'\|_1 = \frac{n}{2}$.
(ii) If (n, m, S, d) is a no-instance, then $\|P'\|_1 \leq \frac{n}{2} - \delta$.
(iii) Moreover, given a vector $v \in \mathbb{R}^n$ with $d_1(v, P') < \delta$ and $\|v\|_1 > \frac{n}{2} - \delta$, we can efficiently compute a witness for (n, m, S, d).

Assume the real protocol has adversary-polytope P' as in Lemma 11, and the adversary-polytope of the ideal protocol is the cross-polytope $X = \{x : \|x\|_1 \leq \frac{n}{2} - \frac{\delta}{2}\}$. Then if (n, m, S, d) is a yes-instance, by Lemma 11 there is a $v \in P'$ with $\|v\|_1 = \frac{n}{2}$. Since P' is the adversary-polytope of the real protocol, there exists a strongly δ-good adversary as seen above. Conversely, if A is a good adversary, we have $\langle A, \pi\rangle \in P'$ and $\|\langle A, \pi\rangle\|_1 \geq \frac{n}{2} - \frac{\delta}{2}$. With black-box access to A, we can efficiently sample the distribution $\langle A, \pi\rangle$ with error $\frac{\delta}{2}$ (w.r.t. the l_1-norm). This gives a point v satisfying the conditions of Lemma 11 (iii) and hence yields a witness for (n, m, S, d). The results we achieved so far are summarised as follows:

Lemma 12 *(informal). Assume that we can construct protocols π and ρ such that the adversary-polytope of π is an ε-set cover polytope, and the adversary-polytope of ρ is a cross-polytope (of*

[2] Of course, since $0 \in X$ and 0 does not correspond to a valid probability distribution, the set X cannot be an adversary-polytope. This problem can be solved by down-scaling A and S and embedding them into the subset of $\mathbb{R}^{n+1}$ corresponding to the set of probability distributions. We will ignore this issue in this proof outline and pretend that all points $v \in \mathbb{R}^n$ correspond to valid probability distributions.

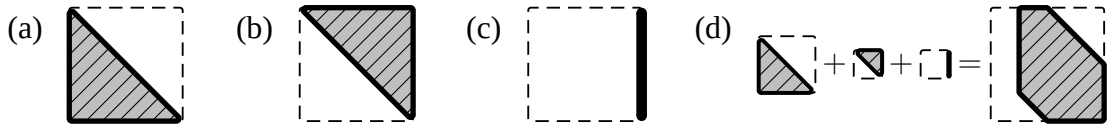

Fig. 1. Polytopes for set cover instance (n, m, S, d) with $n = m = 2$, $d = 1$, $S_1 = \{1, 2\}$, $S_2 = \{1\}$. (a) Polytope D enforcing the condition that the set cover consists of at most one set ($d = 1$). (b) Polytope S^1 enforcing that the set cover contains set S_1 or S_2 (because 1 is contained in both). (c) Polytope S^2 enforcing that the set cover contains S_1 (because 2 is only contained in S_1). (d) Polytope $P_\varepsilon = (1-2\varepsilon)D + \varepsilon S^1 + \varepsilon S^2$ enforces all aforementioned conditions. The only remaining square-vertex is $(1, 0)$, corresponding to the only witness $C = \{1\}$ of (n, m, S, d).

suitable size). Then there is a strongly δ-good adversary if (n, m, S, d) is a yes-instance, and given a strongly good adversary we can efficiently compute a witness for (n, m, S, d).

Constructing the cross-polytope is easy: The cross-polytope X in $\mathbb{R}^n$ has $2n$ vertices $v_1, \ldots, v_{2n}$. We construct the protocol ρ as follows: Upon first activation, ρ expects an $i \in \{1, \ldots, 2n\}$ from the adversary and then chooses its output according to the distribution v_i. By choosing a suitable distribution for i, the adversary can achieve any convex combination of the v_i, so the adversary-polytope is their convex combination X. Since i can be transmitted using $O(\log n)$ bits, the communication complexity of ρ is logarithmic.

Constructing π is more difficult. In general, we cannot expect a set cover polytope to have a polynomial number of vertices, so the approach used for ρ fails. Instead, we have to investigate in more detail which adversary-polytopes can be constructed. First, every polytope consisting of a single point $\{v\}$ can be constructed: the corresponding protocol chooses its output according to the probability distribution v (we call this the singleton-construction). Second, if we can construct the polytopes $P_1, \ldots, P_r$, we can also construct the convex hull P of their union: The corresponding protocol expects an $i \in \{1, \ldots, r\}$ from the adversary and then executes the protocol having adversary-polytope P_i (union-construction). This is a generalisation of the construction of X above. Third, we can also construct $\alpha_1 P_1 + \cdots + \alpha_r P_r$ where $\sum \alpha_i = 1$, $\alpha_i \geq 0$: The protocol randomly chooses an i with probability α_i (sum-construction). In all cases we assume that whenever the protocol makes a random choice, it informs the adversary about the outcome of that choice.

To construct a set cover polytope, we first assume that we are able to construct polytopes for each defining inequality independently. That is, we assume that we can construct the *upper bound polytope* $D := \{v \in [0, 1]^n : \|v\|_1 \leq d\}$ and the *lower bound polytope* $S^i := \{v \in [0, 1]^n : \langle s_{i\cdot}, v\rangle \geq 1\}$. The intersection of these polytopes is P_* which we saw above to be a set cover polytope. Unfortunately, we cannot make use of this fact, since we cannot efficiently construct the intersection as an adversary-polytope. We instead define the *combined polytope* $P_\varepsilon := (1 - m\varepsilon)D + \sum_{i=1}^m \varepsilon S^i$ which can be constructed from D and S^i using the sum-construction. Since there are only $m+1$ summands, the communication complexity is $O(\log m)$. It is left to see that P_ε is an ε-set cover polytope.

Lemma 13. *If $0 < \varepsilon \leq \frac{1}{nm+1}$ then P_ε is an ε-set cover polytope.*

The actual proof is by verifying all inequalities required by Definition 10. For the proof sketch, we instead try to give some geometric motivation (see Figure 1 for an example). First, since the polytopes D and S^i are enclosed in the unit cube $[0,1]^n$, so is P_ε (since the factors in the construction of P_ε add up to 1). Furthermore, let $v \in \{0,1\}^n$ be a cube-vertex that should not be included in P_ε (either because $\|v\|_1 > d$ or because $\langle s_i., v\rangle < 1$). Then in at least one summand R of P_ε (i.e., D or one of the S^i), the cube-vertex v is "cut off" by the inequality defining R. It follows that in the sum P_ε that corresponding vertex is also cut off. Finally, if we choose ε small enough, not too much is cut off, so all cube-vertices that must be contained in P_ε according to Definition 10 are preserved. For a full geometric understanding of the construction, we suggest to examine the example in Figure 1 or the interactive 3D-example in [BU06].

It is left to show that we can construct D and S^i as adversary-polytopes. We will only sketch the construction of D; the polytope S^i is constructed similarly. The vertices of D are $V := \{v \in \{0,1\}^n : \|v\|_1 \leq d\}$. Again, D has an exponential number of vertices, so a direct construction as done for X is not possible. However, each vertex v can be considered as a word of length n and Hamming-weight at most d. If we decompose v into its left and right halves v_l and v_r, we get two words of length $\frac{n}{2}$ and weights d_l, d_2 with $d_l + d_r \leq d$. Thus $V = \bigcup_i V_i \times V_{d-i}$ where i ranges (at most) over $\{0,\dots,d\}$ and V_i is the set of words of length $\frac{n}{2}$ and weight at most i. Since each V_i is again a set of the same structure as V, we can recursively apply that decomposition and construct V from sets of words of length 1. Furthermore, if we again consider V as a subset of $\mathbb{R}^n$, it is $V = \bigcup_i V_i + V_{d-i}$ if we embed the $\frac{n}{2}$-dimensional sets V_i and V_{d-i} suitably into $\mathbb{R}^n$. More exactly, the left summand V_i is embedded into $\mathbb{R}^n$ as $V_i \times \mathbb{R}^{n/2}$ and the right summand V_{d-i} is embedded as $\mathbb{R}^{n/2} \times V_{d-i}$. The recursion is preserved when we take the convex hull, i.e., $\operatorname{conv} V = \bigcup_i \operatorname{conv} V_i + \operatorname{conv} V_{d-i}$. Since $D = \operatorname{conv} V$ we found a recursive construction of D from one-dimensional sets that uses only the unions and sums. The one-dimensional sets have a constant number of vertices and can therefore be directly constructed. The unions can be handled using the union-construction. The sums however cannot be implemented directly. The sum-construction does not allow to construct $\operatorname{conv} V_i + \operatorname{conv} V_{d-i}$, but only $\frac{1}{2}\operatorname{conv} V_i + \operatorname{conv} V_{d-i}$. As a consequence, the resulting polytope is not D, but D scaled by the factor $2^{-O(\log d)}$ where $O(\log d)$ is the depth of the recursion. However, this problem is easily solved by accordingly scaling all other constructions. The communication complexity for realising D is $O(\log n)$ rounds, and $O(\log n)$ communication in each round (the adversary has to choose the index i in the union-construction). This gives communication complexity $O((\log n)^2)$ which is *not* logarithmic. Summarising, we can construct a protocol π with $O((\log n)^2)$ communication complexity that has adversary-polytope P_ε. With Lemma 12 we get:

Lemma 14 *(informal). There are protocols π and ρ with communication complexity $O((\log n)^2)$ such that the following holds: There is a strongly $\Omega(1/\operatorname{poly}(n))$-good adversary if (n, m, S, d) is a yes-instance, and given a strongly good adversary we can efficiently compute a witness for (n, m, S, d).*

The remaining problem is that the communication complexity of π is not logarithmic in n. This can be remedied if we do not require that n is the security parameter. Indeed, if $n := 2^{\sqrt{\log k}}$, then $O((\log n)^2) = O(\log k)$. If we assume that solving NP-complete problems is hard even in

$n^{O(\log n)}$-time, it follows that solving set cover instances with $n := 2^{\sqrt{\log k}}$ is hard in $O(\mathrm{poly}(k))$-time. By construction, the protocols π and ρ share all information with the adversary. From this it can be derived that an adversary is good for (π, ρ) if and only if it is strongly good. Combining these observations with Lemma 14 and the NP-completeness of set cover, we get the following theorem:

Theorem 15. *If* $\mathrm{NP} \not\subseteq \mathrm{BPTIME}(n^{O(\log n)})$*, the following holds for all* $\varepsilon > 0$*: There is no efficient probabilistic algorithm that finds a good adversary for a pair of polynomial-time algorithms with logarithmic communication complexity, even when they are guaranteed to have a strongly* k^{ε}*-good adversary.*

This result already almost separates statistical and computational security for logarithmic protocols. However, two problems still have to be solved. First, a separation not only requires that it is hard to find a good polynomial-time adversary, but that such an adversary does not even exist. Second, an adversary may not be good while still being successful in distinguishing the real and the ideal protocol, because the simulator (which is also computationally bounded) does not simulate optimally. The second problem can be solved by showing that at least for the protocols constructed here, there exists an efficient black-box simulator that simulates *perfectly* if the adversary is not strongly good. To solve the first problem, however, we have to strengthen our assumption:

Assumption 16. *There exists a sequence* f_n *of Boolean formulas computable in deterministic polynomial time such that infinitely many* f_n *are satisfiable and such that for any probabilistic Turing machine* A *that runs in* $n^{O(\log n)}$*-time, the probability* $\Pr[f_n(A(1^n)) = 1]$ *is negligible in* n*.*

We now construct protocols $\tilde{\pi}$ and $\tilde{\rho}$ which on input the security parameter k compute $f_{2^{\sqrt{\log k}}}$, convert it into a set cover instance and then run π or ρ, respectively, on this instance. As infinitely many f_n are satisfiable, strongly $\Omega(1/\mathrm{poly}(2^{\sqrt{\log k}}))$-good adversaries exist infinitely often by Lemma 14. So $\tilde{\pi}$ is not statistically as secure as $\tilde{\rho}$. However, if some polynomial-time adversary was good for infinitely many k, we could use Lemma 14 to find witnesses for f_n with non-negligible probability in n. This yields the following theorem:

Theorem 17 (Computational Does Not Imply Statistical Stand-Alone Security Without Auxiliary Input). *If Assumption 16 holds, computational stand-alone security without auxiliary input does not imply statistical stand-alone security without auxiliary input for polynomial-time protocols with logarithmic communication complexity.*

It is easy to see that this result also holds in the case with *uniform* auxiliary input (in the sense of [Gol93]). However, Theorem 17 does not cover the case with *nonuniform* auxiliary input. This reason is that Assumption 16 cannot hold for nonuniform adversaries. In fact, if we allow a nonuniform input it turns out that whenever a good (but potentially unbounded) adversary exists, its strategy can be encoded into the auxiliary input. For details, see Appendix D.3. This yields the following result:

Theorem 18 (Computational Implies Statistical Stand-Alone Security With Nonuniform Auxiliary Input). *Let* π *and* ρ *be polynomial-time ITMs such that the communication complexity*

and the length of the output of π and ρ on input $(1^k, z)$ is logarithmic in k. If π is as secure as ρ with respect to computational stand-alone security with auxiliary input, then π is as secure as ρ with respect to statistical stand-alone security with auxiliary input.

6 Advantage-Based Security

In the case of advantage-based security, we show that statistical and computational security coincide. The basic idea of our proof is as follows. A protocol B as in Definition 3 can be considered as a one-player-game G^B, the adversary A being the player. The payoff of the game is the output of B. Then the expected payoff for a given adversary A is the advantage $\Pr[\langle A, B\rangle = 1]$. Thus an optimal strategy for the game G^B corresponds to an adversary with maximal advantage. If we can show that a nearly optimal strategy for G^B can be found in polynomial time, it follows that for any successful adversary, there is a successful polynomial-time adversary, and thus statistical and computational security coincide.

Two obstacles have to be overcome. First, in the advantage-based security definition, B has an input, while in the game-theoretic setting, the concept of an external input to the game does not exist. However, when inspecting the definition of advantage-based security, we see that the input x is chosen jointly with the adversary, so we can assume it to be chosen by the adversary. Since we assume a logarithmic bound on B's communication complexity, there is a polynomial n such that the length of x is bounded by $\log n$. Moreover, we deal with a sequence of games, parametrised by the security parameter, giving rise to the following definition:

Definition 19 (Game of a Protocol). *Let B be an ITM. The* game $G^B_{k,n}$ of the protocol B *is the following one-player game:*

- *First, player 1 may choose a string x with $|x| \leq \log n$.*
- *Then, the game consists of the interaction $\langle A, B(1^k, x)\rangle$, where player 1 learns all messages that A receives, and may choose all message that A sends.*
- *The payoff of the game is 1 if B outputs 1, and 0 otherwise.*

This of course does not yield a one-to-one correspondence between optimal adversaries and (sequences of) optimal strategies anymore. A strategy μ incorporates an input x, while the corresponding adversary A^G only implements the behaviour *after* choosing x. Nevertheless, for an adversary A^G corresponding to an optimal strategy, we get

$$\max_{|x|\leq \log n} \Pr[\langle A^G(1^k), B(1^k, x)\rangle] \geq \max_{A, |x|\leq \log n} \Pr[\langle A(1^k), B(1^k, x)\rangle]$$

since the maximum ranges over all x, in particular over the one that μ would have chosen. (Here we use that μ can be assumed to be deterministic.) Since B has logarithmic communication complexity, the game tree of $G^B_{k,n}$ has polynomial size. For one-player-games optimal strategies can be found in polynomial-time in the size of the game tree, yielding the following result (both in the case with and without auxiliary input):

Lemma 20 *(informal). If we can efficiently compute the game tree of $G^B_{k,n}$, there is an optimal polynomial-time adversary. Hence computational and statistical advantage-based security coincide.*

The second obstacle is the fact that in general we cannot efficiently compute the game tree of $G^B_{k,n}$. We remedy this problem by sampling the probabilities in the game tree yielding an approximation. If μ is an optimal strategy for the approximated game, then the expected payoff of μ in the *original* game is at most $\frac{1}{p}$ below the optimum where p is a polynomial we may choose. If B is statistically γ-insecure, there is an adversary A such that (omitting arguments) $\Pr[\langle A, B\rangle] \geq \gamma + \frac{1}{q}$ infinitely often for some polynomial q. By chosing e.g., $p := 2q$, it follows that $\Pr[\langle A^G, B\rangle] \geq \gamma + \frac{1}{2q}$. Since A^G runs in polynomial time, computational γ-insecurity of B follows. Concluding, we have the following result:

Theorem 21 (Computational Implies Statistical Advantage-Based Security). *Let B be a polynomial-time ITM that upon input $(1^k, x)$ has logarithmic communication complexity in k and reads only a prefix of x of logarithmic length in k. Assume that B is γ-secure for some function γ with respect to computational advantage-based security without auxiliary input. Then B is γ-secure with respect to statistical advantage-based security without auxiliary input The same holds for advantage-based security with auxiliary input.*

A Correspondence Between Main Part and Appendix

To make the appendix more readable, we have repeated most of the definitions and theorems from the main part of this paper in the appendix (sometimes in greater detail). To make it easier to find details and proofs for a definitions or theorem in the main part of the paper, we give the correspondences between the main part and the appendix in the following table.

Main part	**Appendix**
Definition 1	Definition 24 on page 17
Definition 2	Definition 31 on page 27
Definition 3	Definition 61 on page 47
Theorem 4	Theorem 23 on page 15
Lemma 5	Lemma 27 on page 24
Theorem 6	Theorems 29 and 30 on pages 24 and 26, resp.
Definition 7	Definition 35 on page 28
Definition 8	Definition 34 on page 28
Definition 9	Definition 33 on page 28
Definition 10	Definition 36 on page 29
Lemma 11	Lemma 37 on page 29
Lemma 12	No exact correspondence. Implicit in the proof of Theorem 49 on 38
Lemma 13	Lemma 39 on page 30
Lemma 14	Theorem 49 on page 38
Theorem 15	Corollary 51 on page 40
Assumption 16	Assumption 52 on page 41
Theorem 17	Theorem 58 on page 44
Theorem 18	Theorem 60 on page 46
Definition 19	Definition 62 on page 47
Lemma 20	No exact correspondence. Implicitly contained in the proof of Theorem 69 on page 52
Theorem 21	Theorem 69 on page 52

B Indistinguishability of Logarithmic Random Variables – Details and Proofs

Before we can prove that for efficiently sampleable random variables of logarithmic length computational and statistical indistinguishability coincide, we first need the following lemma that states that the distributions of such random variables can be estimated sufficiently well.

Lemma 22 (Estimation of Random Variables). *Let $Z \subseteq \{0,1\}^*$. Let $X = \{X_z\}_{z \in Z}$ be an efficiently constructible family of random variables of logarithmic length in $|z|$.*

Then there exists a probabilistic polynomial-time algorithm S_X with the following property:

Upon input $(z, 1^f)$, the algorithm S_X outputs the description of a probability distribution $\tilde{X}$, with the property that $\Delta(X_z; \tilde{X}) \leq \frac{1}{f}$ holds with probability at least $1 - \frac{1}{f}$.

Proof. Let $l(k) \geq 1$ be an efficiently computable logarithmic bound on the length of X_z for all $|z| = k$. Let M_k be the set of all strings of length at most $l(k)$. (Then we always have $X_z \in M_{|z|}$.) Note that $\#M_k$ is polynomially bounded in k.

We defined the algorithm S_X as follows: On input $(z, 1^f)$, let $n := \frac{1}{16} \cdot \#M_{|z|}^3 \cdot f^3$ and choose independent values $x_1, \ldots, x_n$ distributed according to X_z. Let $P_x := \#\{i \leq n : x_i = x\}/n$ be the relative frequency of x in our sample. Output the probabilities $\{P_x\}_{x \in M_{|z|}}$ as rational numbers. (I.e., the P_x define a distribution $\tilde{X}$ with $\Pr[\tilde{X} = x] = P_x$.)

Obviously, we have $\sum_x P_x = 1$, thus $\tilde{X}$ is a probability distribution.

Fix some $z \in Z$ and $f \in \mathbb{N}$. Since $n \cdot P_x$ has (n, p)-binomial distribution for $p := \Pr[X_z = x]$, we have $\mathrm{E}[nP_x] = np$ and $\mathrm{Var}[nP_x] = np(1-p) \leq \frac{n}{4}$. Hence $\mathrm{E}[P_x] = p$ and $\mathrm{Var}[P_x] \leq \frac{1}{4n}$. From this it follows that for any $z \in Z$, it holds that

$$\begin{aligned}
&\Pr\big[\exists x \in M_{|z|} : |P_x - \Pr[X_z = x]| > \tfrac{2}{f \cdot \#M_{|z|}}\big] \\
&\qquad \leq \sum_{x \in M_{|z|}} \Pr\big[|P_x - \Pr[X_z = x]| > \tfrac{2}{f \cdot \#M_{|z|}}\big] \\
&\qquad \leq \sum_{x \in M_{|z|}} \Pr\left[\big|P_x - \mathrm{E}[P_x]\big| \geq \frac{4\sqrt{n}}{f \cdot \#M_{|z|}} \cdot \sqrt{\mathrm{Var}[P_x]}\right] \\
&\qquad \overset{(*)}{\leq} \sum_{x \in M_{|z|}} \frac{f^2 \cdot \#M_{|z|}^2}{16n} = \frac{f^2 \cdot \#M_{|z|}^3}{16n} = \frac{1}{f}.
\end{aligned}$$

Here $(*)$ is an application of Chebyshev's inequality.

Therefore the following holds with probability at least $1 - \frac{1}{f}$:

$$\forall x \in M_{|z|} : |P_x - \Pr[X_z = x]| \leq \frac{2}{f \cdot \#M_{|z|}}. \tag{1}$$

If (1) holds, we have

$$\Delta(\tilde{X}; X_z) = \tfrac{1}{2} \sum_{x \in M_{|z|}} |P_x - \Pr[X_z = x]| \leq \tfrac{1}{2} \cdot \#M_{|z|} \cdot \frac{2}{f \cdot \#M_{|z|}} = \frac{1}{f}.$$

Since (1) holds with probability at least $1 - \frac{1}{f}$, the lemma follows. □

We can now prove that for efficiently sampleable random variables of logarithmic length computational and statistical indistinguishability coincide.

Theorem 23 (Indistinguishability of Logarithmic Random Variables). *Let $Z \subseteq \{0,1\}^*$. Let $X = \{X_z\}_{z \in Z}$ and $Y = \{Y_z\}_{z \in Z}$ be efficiently constructible families of random variables of logarithmic length.*

If X and Y are computationally indistinguishable, then they are statistically indistinguishable.

Proof. Assume that X and Y are computationally indistinguishable.

Let S_X and S_Y be algorithms as in Lemma 22. We define a probabilistic polynomial-time algorithm D as follows: On input $(z, 1^f, x)$, invoke $d_X \leftarrow S_X(z, 1^f)$ and $d_Y \leftarrow S_Y(z, 1^f)$. Then d_X and d_Y are the descriptions of some distributions $\tilde{X}$ and $\tilde{Y}$. If $\Pr[\tilde{X} = x] \geq \Pr[\tilde{X} = y]$, return 1, otherwise 0.

For $k, f \in \mathbb{N}$, let

$$\Delta_f(k) := \max_{\substack{z \in Z \\ |z|=k}} \left| \Pr[D(z, 1^{f(k)}, X_z) = 1] - \Pr[D(z, 1^{f(k)}, Y_z) = 1] \right|.$$

We also define Δ_f for functions f by $\Delta_f(k) := \Delta_{f(k)}(k)$.

First we are going to show that for any function f, we have

$$\Delta(X_z; Y_z) \leq \Delta_f(|z|) + \frac{6}{f(|z|)}. \tag{2}$$

For fixed d_X and d_Y, let $D^*(x) := 1$ if $\Pr[\tilde{X} = x] \geq \Pr[\tilde{Y} = x]$, and $D^*(x) := 0$ otherwise.

First, fix some $z \in Z$. Assume that some d_X and d_Y are given with $\Delta(\tilde{X}; X_z) \leq \frac{1}{f(|z|)}$ and $\Delta(\tilde{Y}; Y_z) \leq \frac{1}{f(|z|)}$. We then have

$$\Delta(\tilde{X}; \tilde{Y}) = \tfrac{1}{2} \sum_x \left| \Pr[\tilde{X} = x] - \Pr[\tilde{Y} = x] \right| = \Pr[D^*(\tilde{X}) = 1] - \Pr[D^*(\tilde{Y}) = 1]. \tag{3}$$

However, we also have $\left| \Pr[D^*(\tilde{X}) = 1] - \Pr[D^*(X_z) = 1] \right| \leq \Delta(\tilde{X}; X_z) \leq \frac{1}{f(|z|)}$, and analogously for $\tilde{Y}$ and Y_z. By the triangle inequality we have $\Delta(X_z; Y_z) \leq \Delta(X_z; \tilde{X}) + \Delta(\tilde{X}; \tilde{Y}) + \Delta(\tilde{Y}; Y_z) \leq \Delta(\tilde{X}; \tilde{Y}) + \frac{2}{f(|z|)}$. Combining these inequalities with (3) we get

$$\Delta(X_z; Y_z) \leq \Pr[D^*(X_z) = 1] - \Pr[D^*(Y_z) = 1] + \frac{4}{f(|z|)}. \tag{4}$$

By construction, $D(z, 1^{f(|z|)}, x)$ first chooses d_X and d_Y using S_X and S_Y, and then outputs $D^*(x)$. We have $\Delta(\tilde{X}; X_z) > \frac{1}{f(|z|)}$ at most with probability $\frac{1}{f(|z|)}$ by definition of S_X, and analogously for $\Delta(\tilde{Y}; Y_z)$. Therefore the conditions under which we showed (4) are fulfilled with probability at least $1 - \frac{2}{f(|z|)}$. Consequently, we have

$$\Delta(X_z; Y_z) \leq \Pr[D(z, 1^{f(|z|)}, X_z)] - \Pr[D(z, 1^{f(|z|)}, X_z)] + \frac{6}{f(|z|)}.$$

This shows (2). In particular, if f is superpolynomial and Δ_f is negligible, then $\Delta(X_z, Y_z)$ is negligible in $|z|$.

For any polynomial p, $D(z, 1^{p(|z|)}, x)$ runs in polynomial time in $|z|$, hence using the computational indistinguishability of X_z and Y_z, it follows that Δ_p is negligible (otherwise $D(z, 1^{p(|z|)}, x)$ would be a distinguisher).

We now show that there is some superpolynomial function f such that Δ_f is negligible. This will show that $\Delta(X_z; Y_z)$ is negligible in $|z|$ and hence conclude the proof.

We say that a function μ^* asymptotically dominates a function μ if for sufficiently large k, we have $\mu^* \geq \mu$. In [Bel02] it is shown that for any countable set N of negligible functions, there exists a negligible function μ^* such that the function μ^* asymptotically dominates μ for any $\mu \in N$.

Let P be the set of all positive polynomials with integer coefficients. Then P is countable, so there exists a function μ^* such that for any $p \in P$, the function μ^* asymptotically dominates Δ_p.

Let $f(k) := \max\{f \in \mathbb{N} : \Delta_f(k) \leq \mu^*\}$. Then $\Delta_f \leq \mu^*$ and therefore Δ_f is negligible. Further, we show that f is superpolynomial. For contradiction, assume that f is not superpolynomial. Then there exists a polynomial $p \in P$ such that $f(k) < p(k)$ for infinitely many k. Then, we also have $\Delta_p(k) > \mu^*(k)$ for infinitely many k (by construction of f). This is a contradiction to the fact that μ^* asymptotically dominates Δ_p. Therefore f is superpolynomial.

In a nutshell, there is a superpolynomial function f such that Δ_f is negligible, and by (2) we have $\Delta(X_z; Y_z) \leq \Delta_f(|z|) + \frac{1}{f(|z|)}$, so X_z and Y_z are statistically indistinguishable. □

C Security with Environment – Details and Proofs

We first give definitional sketches of two popular variants of security with environment: Reactive Simulatability (RSIM) and Universal Composability (UC). Since the full definitions of the underlying machine model and network semantics, we refer the reader to [BPW04] for the RSIM model and [Can05] for the UC model.

Definition 24 (Reactive Simulatability (sketch)). *A protocol π is as secure as a protocol ρ with respect to* computational general reactive simulatability *if for every polynomial-time machine A (the adversary) and every polynomial-time machine H (the honest user) there is a polynomial-time machine S (the simulator) such that*

$$\left\{view_{\pi,A,H,k}(H)\right\}_{k\in\mathbb{N}} \qquad \textit{and} \qquad \left\{view_{\rho,S,H,k}(H)\right\}_{k\in\mathbb{N}}$$

are computationally indistinguishable.

A protocol π is as secure as a protocol ρ with respect to computational universal reactive simulatability *if for every polynomial-time machine A (the adversary) there is a polynomial-time machine S (the simulator) such that for every polynomial-time machine H (the honest user)*

$$\left\{view_{\pi,A,H,k}(H)\right\}_{k\in\mathbb{N}} \qquad \textit{and} \qquad \left\{view_{\rho,S,H,k}(H)\right\}_{k\in\mathbb{N}}$$

are computationally indistinguishable in k.

We speak about statistical *general/universal reactive simulatability if in the above definitions A, H and S are unbounded and statistical indistinguishability is used instead of computational indistinguishability.*

Definition 25 (Universal Composability (sketch)). *A protocol π is as secure as a protocol ρ with respect to* computational UC, *if for every polynomial-time machine A (the adversary) there*

is a polynomial-time machine S (the simulator) such that for every polynomial-time machine Z (the environment) and every sequence z of strings of polynomial length,

$$\left\{EXEC_{\pi,A,Z}(k,z_k)\right\}_{k\in\mathbb{N}} \qquad \textit{and} \qquad \left\{EXEC_{\rho,S,Z}(k,z_k)\right\}_{k\in\mathbb{N}}$$

are computationally indistinguishable.

We speak about statistical UC *if in the above definition A, Z and S are unbounded and statistical indistinguishability is used instead of computational indistinguishability.*

In this definition, we assumed that the output of the environment may be a string. Another variant of UC that is often considered requires the environment to give a single bit as output. These variants are equivalent [Can05, Section 4.3, "On environments with non-binary outputs"]. The definition of statistical UC is sketched in [Can05, Section 4.2, "On statistical and perfect emulation"].

In order to capture all the above definitions of security with environment, we take a generalised point of view that can capture both settings. For this, we consider the execution of the real protocol π (including the adversary) as an oracle Turing machine X that takes the environment/honest user as an oracle and outputs its view or output, respectively. Similarly, an oracle Turing machine Y represents the ideal protocol ρ together with the simulator. Thus we can first analyse the security of logarithmic protocols in an exact and simple setting in Lemma 26, and then derive results for the more conventional settings of RSIM and UC in Theorems 29 and 30, respectively.

Given two oracle Turing machines X and Y, we say that all oracle queries and oracle answers can be extracted from the output of X and Y if the following holds: For every n, there is a function f_n such that for any oracle $\mathcal{O}$, the following two conditions are fulfilled: (i) We have $f_n(X^{\mathcal{O}}) = (i,o)$ where i and o are the input and output of $\mathcal{O}$ in the n-th oracle query in an execution of $X^{\mathcal{O}}$. (ii) We have $f_n(Y^{\mathcal{O}}) = (i,o)$ where i and o are the input and output of $\mathcal{O}$ in the n-th oracle query in an execution of $Y^{\mathcal{O}}$.

Lemma 26. *Let X and Y be oracle Turing machines. Let A be an oracle. Assume both X and Y call their oracle at most r times, and that the total length of the answers given by A is at most l. Assume further that all oracle queries and oracle answers can be extracted from the output of X and Y.*

Let $\mathcal{D}$ be some distribution on the set of r-tupels of strings. Let $\tilde{A}$ be the oracle that chooses an r-tupel $(o_1,\ldots,o_r)$ of strings according to $\mathcal{D}$ and in its i-th activation responds with o_i.

Let $\mathfrak{O} \subseteq (\{0,1\}^*)^r$ be the set of all r-tupels $\mathbf{w}$ satisfying that the total length $\sum_{i=1}^{r} \mathbf{w}_i$ is at most l. Let $p_{\min} := \min_{\mathbf{w}\in\mathfrak{O}} \Pr_{\mathcal{D}}[\mathbf{w}]$.

Then $\Delta(X^{\tilde{A}}; Y^{\tilde{A}}) \geq 3^{-r} p_{\min}\, \Delta(X^A; Y^A)$.

Proof. If $\Pr_{\mathcal{D}}[\mathbf{w}] = 0$ for some $\mathbf{w} \in \mathfrak{O}$, we have $p_{\min} = 0$ and the lemma is trivially fulfilled. We can therefore assume $\Pr_{\mathcal{D}}[\mathbf{w}] \neq 0$ for all $\mathbf{w} \in \mathfrak{O}$.

To show the lemma, we first define some random variables. In an execution of X^A, let X denote the output of X^A, let I_n^X denote the input to the oracle A in the n-th query, and let O_n^X denote the corresponding response of A (with $I_n^X = O_n^X = \perp$ if A is queried less than n times). Let $V_n^X := (I_1^X, O_1^X, \ldots, I_n^X, O_n^X)$. Then $V^X := V_r^X$ is the view of A.

Analogously, we define the random variables Y, I_n^Y, O_n^Y, V_n^Y and V^Y for an execution of Y^A.

For executions of $X^{\tilde{A}}$ and $Y^{\tilde{A}}$ we augment the random variables with a tilde (e.g., $\tilde{O}_n^Y$ is the n-th output of $\tilde{A}$ in an execution of $Y^{\tilde{A}}$).

In the following we use the convention that $0 \cdot \Pr[A|C] = 0$, even if $\Pr[C] = 0$ (and thus $\Pr[A|C]$ is undefined). Similarly, we let $0 \cdot \Delta(A|C; B|D) = 0$ even if $\Pr[C] = 0$ or $\Pr[D] = 0$. The main effect of this convention is that the Bayesian rule $\Pr[A, C] = \Pr[A|C] \cdot \Pr[C]$ holds even if $\Pr[C] = 0$.

For any finite sequence $\mathbf{w}$ of strings, let $\#\mathbf{w}$ denote the number of elements of $\mathbf{w}$, and $\|\mathbf{w}\|$ the total length of the strings. That is, if $\mathbf{w} = (o_1, \ldots, o_n)$, we have $\#\mathbf{w} = n$ and $\|\mathbf{w}\| = \sum_{i=1}^{n} |o_i|$.

Let o be a string. If $\#\mathbf{w} < r$ and $\|\mathbf{w}\| + |o| \leq l$, let $p(o|\mathbf{w}) := \Pr[\mathbf{W}_{\#\mathbf{w}+1} = o | (\mathbf{W}_1, \ldots, \mathbf{W}_{\#\mathbf{w}}) = \mathbf{w}]$ where $\mathbf{W}$ is a random variable distributed according to $\mathcal{D}$.

For $\#\mathbf{w} = r$ and $\|\mathbf{w}\| \leq l$, let $\alpha_{\mathbf{w}} := 1$. For $\#\mathbf{w} < r$ and $\|\mathbf{w}\| \leq l$, let

$$\alpha_{\mathbf{w}} := \tfrac{1}{3} \min_o p(o|\mathbf{w}) \cdot \alpha_{\mathbf{w}\|o}$$

where the minimum ranges over all strings o with $\|\mathbf{w}\| + |o| \leq l$. Here $\mathbf{w}\|o$ denotes the result of appending the element o to the sequence $\mathbf{w}$.

By induction, it follows that

$$\alpha_\lambda = 3^{-r} \min_{\substack{\#\mathbf{w}=r \\ \|\mathbf{w}\|=l}} \prod_{i=1}^{r} p\big(\mathbf{w}_i | (\mathbf{w}_1, \ldots, \mathbf{w}_{i-1})\big) = 3^{-r} \min_{\substack{\#\mathbf{w}=r \\ \|\mathbf{w}\|=l}} P(\mathbf{W} = r) = 3^{-r} p_{\min}$$

where $\mathbf{W}$ is again distributed according to $\mathcal{D}$. Here λ denotes the empty sequence.

For some (partial) view $v = (i_1, o_1, \ldots, i_n, o_n)$, let $\mathbf{w}(v) := (o'_1, \ldots, o'_n)$ where $o'_i := \lambda$ if $o_i = \perp$, and $o'_i := o_i$ otherwise (i.e., $\mathbf{w}(v)$ denotes the sequence of the outputs of A or $\tilde{A}$ in the view v, where we assume the empty output λ for the i-th query if there was no i-th query).

Let

$$\mathcal{V}_n := \big\{ v : \Pr[V_n^X = v] > 0,\ \Pr[V_n^Y = v] > 0,\ \Pr[\tilde{V}_n^X = v] > 0,\ \Pr[\tilde{V}_n^Y = v] > 0 \big\}$$

and

$$\begin{aligned} \mathcal{VI}_n := \big\{ (v, i) : \Pr[V_{n-1}^X = v, I_n^X = i] > 0, \quad & \Pr[V_{n-1}^Y = v, I_n^Y = i] > 0, \\ \Pr[\tilde{V}_{n-1}^X = v, \tilde{I}_n^X = i] > 0, \quad & \Pr[\tilde{V}_{n-1}^Y = v, \tilde{I}_n^Y = i] > 0 \big\}. \end{aligned}$$

For any $v \in \mathcal{V}_r$ and all x, it is $P(X = x | V^X = v) = P(\tilde{X} = x | \tilde{V}^X = v)$. The same holds for Y instead of X. So for all $v \in \mathcal{V}_r$, it is

$$\Delta(X|V^X = v;\ Y|V^Y = v) = \Delta(\tilde{X}|\tilde{V}^X = v;\ \tilde{Y}|\tilde{V}^Y = v). \tag{5}$$

Here and in the following $A|B$ denotes the random variable A conditioned on the event B.

Now fix some $1 \leq n \leq r$ and assume that

$$\Delta(\tilde{X}|\tilde{V}_n^X = v';\ \tilde{Y}|\tilde{V}_n^Y = v') \geq \alpha_{\mathbf{w}(v')} \Delta(X|V_n^Y = v';\ Y|V_n^Y = v') \tag{6}$$

for all $v' \in \mathcal{V}_n$.

We try to bound $\Delta(\tilde{X}|\tilde{V}^X_{n-1} = v, \tilde{I}^X_n = i;\ \tilde{Y}|\tilde{V}^Y_{n-1} = v, \tilde{I}^Y_n = i)$ from below for all $(v,i) \in \mathcal{VI}_n$. First, we find that

$$\begin{aligned}
&\Delta(X|V^X_{n-1} = v, I^X_n = i; \quad Y|V^Y_{n-1} = v, I^Y_n = i) \\
&\overset{\text{(i)}}{=} \Delta(X, O^X_n|V^X_{n-1} = v, I^X_n = i; \quad Y, O^Y_n|V^Y_{n-1} = v, I^Y_n = i) \\
&= \tfrac{1}{2}\sum_{o,x}\Big|\Pr[X = x|V^X_{n-1} = v, I^X_n = i, O^X_n = o] \cdot \Pr[O^X_n = o|V^X_{n-1} = v, I^X_n = i] \\
&\qquad - \Pr[Y = x|V^Y_{n-1} = v, I^Y_n = i, O^Y_n = o] \cdot \Pr[O^Y_n = o|V^Y_{n-1} = v, I^Y_n = i]\Big| \\
&\overset{\text{(ii)}}{=} \sum_o \Pr[O^X_n = o|V^X_{n-1} = v, I^X_n = i] \\
&\qquad \cdot \Delta\big(X|V^X_{n-1} = v, I^X_n = i, O^X_n = o; \quad Y|V^Y_{n-1} = v, I^Y_n = i, O^Y_n = o\big) \qquad (7)
\end{aligned}$$

Here (i) stems from the fact that by assumption, the oracle responses and thus in particular O^X_n and O^Y_n can be extracted from X and Y, respectively. We have (ii) because $\Pr[O^X_n = o|V^X_{n-1} = v, I^X_n = i] = \Pr[O^Y_n = o|V^Y_{n-1} = v, I^Y_n = i]$ (which again holds because the n-th oracle answer depends only on the oracle and its view so far).

From (7) we get that there is some $\hat{o}$ (depending on i and v) such that the following three inequalities hold:

$$\Pr[V^X_{n-1} = v, I^X_n = i, O^X_n = \hat{o}] > 0, \quad \Pr[V^Y_{n-1} = v, I^Y_n = i, O^Y_n = \hat{o}] > 0, \qquad (8)$$

and

$$\begin{aligned}
&\Delta\big(X|V^X_{n-1} = v, I^X_n = i, O^X_n = \hat{o};\ Y|V^Y_{n-1} = v, I^Y_n = i, O^Y_n = \hat{o}\big) \\
&\qquad\qquad \geq \Delta(X|V^X_{n-1} = v, I^X_n = i;\ Y|V^Y_{n-1} = v, I^Y_n = i). \qquad (9)
\end{aligned}$$

Since the total length of all query answers given by A is bounded by l by assumption, it is $\|\mathbf{w}(v)\| + |\hat{o}| \leq l$ or $\hat{o} = \perp$.

Since $\hat{o} = \perp$ only if $i = \perp$, and since $(v,i) \in \mathcal{VI}_n$, and using the fact that $\tilde{A}$ when activated outputs any string $\hat{o}$ with $\|\mathbf{w}(v)\| + |\hat{o}| \leq l$ with nonzero probability, we further have $\Pr[\tilde{V}^X_{n-1} = v, \tilde{I}^X_n = i, \tilde{O}^X_n = \hat{o}] > 0$ and $\Pr[\tilde{V}^Y_{n-1} = v, \tilde{I}^Y_n = i, \tilde{O}^Y_n = \hat{o}] > 0$. Combining this with (8) we get $v\|(i,\hat{o}) \in \mathcal{V}_n$. (Here $v\|(i,\hat{o})$ denotes the view resulting from appending $(i,\hat{o})$ to v.)

So in the case $i \neq \perp$ we have

$$
\begin{aligned}
&\Delta(\tilde{X}|\tilde{V}^X_{n-1} = v, \tilde{I}^X_n = i; \quad \tilde{Y}|\tilde{V}^Y_{n-1} = v, \tilde{I}^Y_n = i) \\
&\overset{\text{(i)}}{=} \sum_o \Pr[\tilde{O}^X_n = o|\tilde{V}^X_{n-1} = v, \tilde{I}^X_n = i] \\
&\qquad \cdot \Delta\big(\tilde{X}|\tilde{V}^X_{n-1} = v, \tilde{I}^X_n = i, \tilde{O}^X_n = o; \quad \tilde{Y}|\tilde{V}^Y_{n-1} = v, \tilde{I}^Y_n = i, \tilde{O}^Y_n = o\big) \\
&\overset{\text{(ii)}}{=} \sum_{\substack{o \text{ with} \\ \|\mathbf{w}(v)\|+|o| \leq l}} p(o|\mathbf{w}(v))\, \Delta\big(\tilde{X}|\tilde{V}^X_{n-1} = v, \tilde{I}^X_n = i, \tilde{O}^X_n = o; \quad \tilde{Y}|\tilde{V}^Y_{n-1} = v, \tilde{I}^Y_n = i, \tilde{O}^Y_n = o\big) \\
&\geq p(\hat{o}|\mathbf{w}(v))\, \Delta\big(\tilde{X}|\tilde{V}^X_{n-1} = v, \tilde{I}^X_n = i, \tilde{O}^X_n = \hat{o}; \quad \tilde{Y}|\tilde{V}^Y_{n-1} = v, \tilde{I}^Y_n = i, \tilde{O}^Y_n = \hat{o}\big) \\
&\overset{(6)}{\geq} p(\hat{o}|\mathbf{w}(v))\, \alpha_{(\mathbf{w}(v),\hat{o})}\, \Delta\big(X|V^X_{n-1} = v, I^X_n = i, O^X_n = \hat{o}; \quad Y|V^Y_{n-1} = v, I^Y_n = i, O^Y_n = \hat{o}\big) \\
&\geq 3\alpha_{\mathbf{w}(v)}\, \Delta\big(X|V^X_{n-1} = v, I^X_n = i, O^X_n = \hat{o}; \quad Y|V^Y_{n-1} = v, I^Y_n = i, O^Y_n = \hat{o}\big) \\
&\overset{(9)}{\geq} 3\alpha_{\mathbf{w}(v)}\, \Delta\big(X|V^X_{n-1} = v, I^X_n = i; \quad Y|V^Y_{n-1} = v, I^Y_n = i\big). \qquad (10)
\end{aligned}
$$

Here equality (i) is proven exactly like (7), and (ii) uses the fact tht $\tilde{A}$'s answers are distributed according to $\mathcal{D}$ by construction. At this point, we used that $i \neq \perp$, since $\tilde{I}^X_n = \perp$ means that there is no n-th oracle query and therefore $\tilde{O}^X_n = \perp$.

In the case $i = \perp$, i.e., in the case where no n-th oracle query occurs, from $\tilde{I}^X_n = i$ it follows that $\tilde{O}^X_n = i$ (and the same for Y), so we have

$$
\begin{aligned}
&\Delta(\tilde{X}|\tilde{V}^X_{n-1} = v, \tilde{I}^X_n = i; \quad \tilde{Y}|\tilde{V}^Y_{n-1} = v, \tilde{I}^Y_n = i) \\
&= \Delta(\tilde{X}|\tilde{V}^X_{n-1} = v, \tilde{I}^X_n = i, \tilde{O}^X_n = \perp; \quad \tilde{Y}|\tilde{V}^Y_{n-1} = v, \tilde{I}^Y_n = i, \tilde{O}^Y_n = \perp) \\
&\overset{(6)}{\geq} \alpha_{(\mathbf{w}(v),\lambda)} \Delta(X|V^X_{n-1} = v, I^X_n = i, O^X_n = \perp; \quad Y|V^Y_{n-1} = v, I^Y_n = i, O^Y_n = \perp) \\
&\geq 3\alpha_{\mathbf{w}(v)} \Delta(X|V^X_{n-1} = v, I^X_n = i, O^X_n = \perp; \quad Y|V^Y_{n-1} = v, I^Y_n = i, O^Y_n = \perp) \\
&= 3\alpha_{\mathbf{w}(v)} \Delta(X|V^X_{n-1} = v, I^X_n = i; \quad Y|V^Y_{n-1} = v, I^Y_n = i)
\end{aligned}
$$

So (10) holds in all cases.

We now want to bound $\Delta(\tilde{X}|\tilde{V}^X_{n-1} = v;\ \tilde{Y}|\tilde{V}^Y_{n-1} = v)$ from below for all $v \in \mathcal{V}_{n-1}$. It is

$$
\begin{aligned}
&\Delta(\tilde{X}|\tilde{V}^X_{n-1} = v;\ \tilde{Y}|\tilde{V}^Y_{n-1} = v) \\
&\overset{\text{(i)}}{=} \Delta(\tilde{X}, \tilde{I}^X_n|\tilde{V}^X_{n-1} = v;\ \tilde{Y}, \tilde{I}^Y_n|\tilde{V}^Y_{n-1} = v) \\
&= \tfrac{1}{2}\sum_{x,i}\big|\Pr[\tilde{X} = x|\tilde{V}^X_{n-1} = v, \tilde{I}^X_n = i] \cdot \Pr[\tilde{I}^X_n = i|\tilde{V}^X_{n-1} = v] \\
&\qquad - \Pr[\tilde{Y} = x|\tilde{V}^Y_{n-1} = v, \tilde{I}^Y_n = i] \cdot \Pr[\tilde{I}^Y_n = i|\tilde{V}^Y_{n-1} = v]\big| \\
&\geq \tfrac{1}{2}\sum_{x,i}\Pr[\tilde{I}^X_n = i|\tilde{V}^X_{n-1} = v] \cdot \big|\Pr[\tilde{X} = x|\tilde{V}^X_{n-1} = v, \tilde{I}^X_n = i] - \Pr[\tilde{Y} = x|\tilde{V}^Y_{n-1} = v, \tilde{I}^Y_n = i]\big| \\
&\qquad - \tfrac{1}{2}\sum_{x,i}\Pr[\tilde{Y} = x|\tilde{V}^Y_{n-1} = v, \tilde{I}^Y_n = i] \cdot \big|\Pr[\tilde{I}^Y_n = i|\tilde{V}^Y_{n-1} = v] - \Pr[\tilde{I}^X_n = i|\tilde{V}^X_{n-1} = v]\big| \\
&= \Big(\sum_i \Pr[\tilde{I}^X_n = i|\tilde{V}^X_{n-1} = v] \cdot \Delta(\tilde{X}|\tilde{V}^X_{n-1} = v, \tilde{I}^X_n = i;\ \tilde{Y}|\tilde{V}^Y_{n-1} = v, \tilde{I}^Y_n = i)\Big) \\
&\qquad\qquad - \Delta(\tilde{I}^X_n|\tilde{V}^X_{n-1} = v;\ \tilde{I}^Y_n|\tilde{V}^Y_{n-1} = v) \qquad (11)
\end{aligned}
$$

Equation (i) uses the fact that the oracle queries, and thus in particular $\tilde{I}^X_n$ and $\tilde{I}^Y_n$ can be extracted from $\tilde{X}$ and $\tilde{Y}$, respectively.

For convenience, we abbreviate $\Delta(\tilde{I}_n^X|\tilde{V}_{n-1}^X = v;\ \tilde{I}_n^Y|\tilde{V}_{n-1}^Y = v)$ as $\Delta_{\tilde{I}}$. Then we continue the above calculation.

$$
\begin{aligned}
(11) &\overset{(10,ii)}{\geq} \Big(\sum_i \Pr[I_n^X = i|V_{n-1}^X = v] \cdot 3\alpha_{\mathbf{w}(v)} \Delta(X|V_{n-1}^X = v, I_n^X = i;\ Y|V_{n-1}^Y = v, I_n^Y = i)\Big) - \Delta_{\tilde{I}} \\
&= \Big(\tfrac{3\alpha_{\mathbf{w}(v)}}{2} \sum_{x,i} \Big|\Pr[X = x|V_{n-1}^X = v, I_n^X = i] \cdot \Pr[I_n^X = i|V_{n-1}^X = v] \\
&\qquad - \Pr[Y = x|V_{n-1}^Y = v, I_n^Y = i] \cdot \Pr[I_n^X = i|V_{n-1}^X = v]\Big|\Big) - \Delta_{\tilde{I}} \\
&= \Big(\tfrac{3\alpha_{\mathbf{w}(v)}}{2} \sum_{x,i} \Big|\Pr[X = x|V_{n-1}^X = v, I_n^X = i] \cdot \Pr[I_n^X = i|V_{n-1}^X = v] \\
&\qquad - \Pr[Y = x|V_{n-1}^Y = v, I_n^Y = i] \cdot \Pr[I_n^Y = i|V_{n-1}^Y = v] \\
&\qquad - \Pr[Y = x|V_{n-1}^Y = v, I_n^Y = i] \cdot \big(\Pr[I_n^X = i|V_{n-1}^X = v] - \Pr[I_n^Y = i|V_{n-1}^Y = v]\big)\Big|\Big) \\
&\qquad - \Delta_{\tilde{I}} \\
&\geq \Big(\tfrac{3\alpha_{\mathbf{w}(v)}}{2} \sum_{x,i} \Big|\Pr[X = x|V_{n-1}^X = v, I_n^X = i] \cdot \Pr[I_n^X = i|V_{n-1}^X = v] \\
&\qquad - \Pr[Y = x|V_{n-1}^Y = v, I_n^Y = i] \cdot \Pr[I_n^Y = i|V_{n-1}^Y = v]\Big| \\
&\qquad - \Pr[Y = x|V_{n-1}^Y = v, I_n^Y = i] \cdot \Big|\Pr[I_n^X = i|V_{n-1}^X = v] - \Pr[I_n^Y = i|V_{n-1}^Y = v]\Big|\Big) \\
&\qquad - \Delta_{\tilde{I}} \\
&= 3\alpha_{\mathbf{w}(v)} \Delta(X, I_n^X|V_{n-1}^X = v;\ Y, I_n^Y|V_{n-1}^Y = v) - 3\alpha_{\mathbf{w}(v)} \Delta(I_n^X|V_{n-1}^X = v;\ I_n^Y|V_{n-1}^Y = v) - \Delta_{\tilde{I}} \\
&\overset{(iii)}{=} 3\alpha_{\mathbf{w}(v)} \Delta(X, I_n^X|V_{n-1}^X = v;\ Y, I_n^Y|V_{n-1}^Y = v) - (1 + 3\alpha_{\mathbf{w}(v)}) \Delta(\tilde{I}_n^X|\tilde{V}_{n-1}^X = v;\ \tilde{I}_n^Y|\tilde{V}_{n-1}^Y = v) \\
&\overset{(iv)}{\geq} 3\alpha_{\mathbf{w}(v)} \Delta(X|V_{n-1}^X = v;\ Y|V_{n-1}^Y = v) - (1 + 3\alpha_{\mathbf{w}(v)}) \Delta(\tilde{X}|\tilde{V}_{n-1}^X = v;\ \tilde{Y}|\tilde{V}_{n-1}^Y = v).
\end{aligned} \tag{12}
$$

Inequality (ii) uses (besides the bound given in (10)) the fact that given the inputs and responses of all oracle queries up to the $(n-1)$-st query, the input of the n-th query depends only on the querying machine X, but not on the oracle. So given $\tilde{V}_{n-1}^X$ (or $\tilde{V}_{n-1}^X$, resp.), $\tilde{I}_n^X$ and I_n^X have the same distribution. Equality (iii) uses exactly the same fact. In equality (iv) we used that I_n^X and I_n^Y can be extracted from X and Y, respectively, and that $\tilde{I}_n^X$ and $\tilde{I}_n^Y$ can be extracted from $\tilde{X}$ and $\tilde{Y}$, respectively.

Since $\|o(v)\| = n - 1 < r$, we have $\alpha_{o(v)} \leq \frac{1}{3}$. From (12) we then get:

$$
\begin{aligned}
\Delta(\tilde{X}|\tilde{V}_{n-1}^X = v;\ \tilde{Y}|\tilde{V}_{n-1}^Y = v) &\geq \frac{3\alpha_{\mathbf{w}(v)}}{2 + 3\alpha_{\mathbf{w}(v)}} \Delta(X|V_{n-1}^X = v;\ Y|V_{n-1}^Y = v) \\
&\geq \alpha_{\mathbf{w}(v)} \Delta(X|V_{n-1}^X = v;\ Y|V_{n-1}^Y = v).
\end{aligned} \tag{13}
$$

So recapitulating, if for some $1 \leq n \leq r$, (6) holds for all $v' \in \mathcal{V}_n$, we have (13) for all $v \in \mathcal{V}_{n-1}$. By induction over decreasing n (using (5) for the induction basis $n = r$) we see that the following holds for all $v \in \mathcal{V}_0$:

$$
\Delta(\tilde{X}|\tilde{V}_0^X = v;\ \tilde{Y}|\tilde{V}_0^Y = v) \geq \alpha_{\mathbf{w}(v)} \Delta(X|V_0^X = v;\ Y|V_0^Y = v).
$$

Since $V_0^X = V_0^Y = \tilde{V}_0^X = \tilde{V}_0^Y$ is the sequence of length 0 with probability 1, we can rewrite the last inequality as $\Delta(\tilde{X};\tilde{Y}) \geq \alpha_\lambda \Delta(X;Y) = 3^{-r} p_{\min} \Delta(X;Y)$. □

We restate Lemma 26 in an asymptotic setting:

Lemma 27. *Let X and Y be oracle Turing machines. Let A be an oracle. Assume both X and Y call their oracle at most r times, and that the total length of the answers given by A is at most l. Assume further that all oracle queries and oracle answers can be extracted from the output of X and Y.*

Let $\tilde{A}$ be the oracle that first uniformly choose an r-tupel $(o_1, \ldots, o_r)$ of strings such that the total length $\sum o_i$ is at most l, and then upon its i-th activation responds with o_i.

Then $\Delta(X^{\tilde{A}}; Y^{\tilde{A}}) \geq 2^{-O(l+r)} \Delta(X^A; Y^A)$.

Proof. We can encode an r-tupel $(o_1, \ldots, o_r)$ with total length at most l as a string of length $O(l + r)$. Therefore there are at most $2^{O(l+r)}$ such r-tupels. If $\mathcal{D}$ is the uniform distribution on these r-tupels, in the notation of Lemma 26 we have $p_{\min} \in 2^{-O(l+r)}$. So by Lemma 26 we have $\Delta(X^{\tilde{A}}; Y^{\tilde{A}}) \geq 3^{-r} 2^{-O(l+r)} \Delta(X^A; Y^A)$. Since $3^{-r} 2^{-O(l+r)} \subseteq 2^{-O(l+r)}$, the lemma follows. □

To apply Lemma 27 to the setting of RSIM or UC, we need the following well-known fact:

Lemma 28. *The following holds with respect to computational/statistical general/universal reactive simulatability and computational/statistical UC.*

Let π and ρ be protocols with communication complexity bounded b. Then there is an adversary A_{dummy} (the so-called dummy-adversary) with communication complexity $O(b)$ such that π is as secure as ρ if and only if π is as secure as ρ with respect to the dummy-adversary A_{dummy}.

This is easily shown using the so-called dummy-adversary technique. See [Can05, Section 4.3.1, "Security with respect to the dummy adversary"] for an overview.

We can now apply Lemma 27 to the setting of RSIM and conclude that in this setting, computational implies statistical security for logarithmic protocols.

Theorem 29 (Computational Implies Statistical Simulatability). *Let π and ρ be polynomial-time protocols with logarithmic communication complexity. Assume that π is as secure as ρ with respect to computational general reactive simulatability. Then π is as secure as ρ with respect to statistical general reactive simulatability.*

The same holds for universal reactive simulatability.

Proof. By definition, the view $view_{\pi,A,H,k}(H)$ consists of the sequence of all messages sent and received by H together with all internal states of H. Similarly, we define the external view $extview_{\pi,A,H,k}(H)$ to consist only of the messages sent and received by H (without the internal states). Obviously, $extview_{\pi,A,H,k}(H)$ is a function of $view_{\pi,A,H,k}(H)$.

By Lemma 28, there exists a logarithmic bound b_A such that we can w.l.o.g. assume all adversaries to have communication complexity at most b_A. Since a simulator that communicates more than the adversary with the honest user will be trivially distinguished from the adversary,

we can assume the simulator to communicate at most b_A with the honest user. Further, since the communication complexity of the protocol ρ is logarithmically bounded, we can assume the total communication of the simulator to be bounded by b_S. Since there are fixed logarithmic upper bounds on the communication complexity of π, ρ, the adversary and the simulator, we can also assume the honest user's communication complexity to have a fixed bound.

By choosing the same logarithmic bound b for all the entities above, we can assume all simulators, adversaries and honest users, as well as the protocols π and ρ to have communication complexity at most b. This holds for statistical and computational general and universal reactive simulatability. We will implicitly assume this bound b for the rest of this proof.

Assume now that π is as secure as ρ with respect to computational general reactive simulatability. We want to show that this implies that π is as secure as ρ with respect to statistical universal reactive simulatability. Since universal reactive simulatability implies general reactive simulatability, this show the theorem both in the case of general and of universal reactive simulatability.

For any adversary, we can now construct a sequence of oracle Turing machines $X_{A,k}$ so that $X_{A,k}^{H(1^k)}$ simulates the interaction between protocol π, adversary A and honest user H upon security parameter k and then outputs the external view of H. Here we identify honest users (which are machines in the sense of the Reactive Simulatability framework) with oracles in the following natural way: A query to $H(1^k)$ corresponds to an activation of H through an incoming message (or in its capacity as scheduler) upon security parameter k, and outgoing messages sent by H are modelled by the oracle responses. Then $X_{A,k}^{H(1^k)}$ and $extview_{\pi,A,H,k}(H)$ have the same distribution. Since A and π have communication complexity at most b, we can assume that the number of times H is called by $X_{A,k}^{H(1^k)}$ and the total length of the answers given by H is bounded by a bound $b'(k) \in O(b(k))$ (independent of the choice of A).

Similarly, for any simulator S we can construct a sequence of oracle Turing machines $Y_{S,k}$ such that $Y_{S,k}^{H(1^k)}$ and $extview_{\rho,S,H,k}(H)$ have the same distribution. As above, we can bound the number of times H is called by $X_{A,k}^{H(1^k)}$ and the total length of the answers given by H by $b'(k)$.

Note that since $X_{A,k}^{H(1^k)}$ and $Y_{S,k}^{H(1^k)}$ output the external view of H by construction, the sequence of all queries to H and of all its answers is contained in the output of $X_{A,k}^{H(1^k)}$ and $Y_{S,k}^{H(1^k)}$, respectively.

Let than $\tilde{H}$ be the honest user/oracle that chooses randomly a sequence of $b'(k)$ messages of total length at most $b'(k)$.

By Lemma 27, if $\Delta(X_{A,k}^{H(1^k)}; Y_{S,k}^{H(1^k)})$ is not negligible in k, then $\Delta(X_{A,k}^{\tilde{H}(1^k)}; Y_{S,k}^{\tilde{H}(1^k)})$ is not negligible in k, either.

We can now finish our proof by showing that π is as secure as ρ with respect to statistical universal reactive simulatability. Let an adversary A be given. By Lemma 28 we can assume A to be polynomial-time. Then, since π is as secure as ρ with respect to computational general reactive simulatability, there is a polynomial-time simulator S such that $\{extview_{\pi,A,\tilde{H},k}(\tilde{H})\}_k$ and $\{extview_{\rho,S,\tilde{H},k}(\tilde{H})\}_k$ are computationally indistinguishable (we are even guaranteed that the views, not only the external views of H are computationally indistinguishable). Since

$extview_{\pi,A,\tilde{H},k}(\tilde{H})$ and $extview_{\rho,S,\tilde{H},k}(\tilde{H})$ can be efficiently computed (π, ρ, A, $\tilde{H}$ and S are polynomial-time), and since the external view of $\tilde{H}$ has logarithmic length, by Theorem 23 if follows that $\{extview_{\pi,A,\tilde{H},k}(\tilde{H})\}_k$ and $\{extview_{\rho,S,\tilde{H},k}(\tilde{H})\}_k$ are even statistically indistinguishable, i.e.,

$$\Delta\big(extview_{\pi,A,\tilde{H},k}(\tilde{H});\ extview_{\rho,S,\tilde{H},k}(\tilde{H})\big) = \Delta\big(X_{A,k}^{\tilde{H}(1^k)}; Y_{S,k}^{\tilde{H}(1^k)}\big)$$

is negligible. Then for any honest user H (not only polynomial-time ones) we have that $\Delta\big(X_{A,k}^{H(1^k)}; Y_{S,k}^{H(1^k)}\big)$ is negligible. In other words, $\{extview_{\pi,A,H,k}(H)\}_k$ and $\{extview_{\rho,S,H,k}(H)\}_k$ are statistically indistinguishable. Since the distribution of the view can be (inefficiently) computed from the external view (given a specific honest user H), it follows that also $\{view_{\pi,A,H,k}(H)\}_k$ and $\{view_{\rho,S,H,k}(H)\}_k$ are statistically indistinguishable.

So π is as secure as ρ with respect to statistical universal reactive simulatability. □

Similar to Theorem 29 we get that also in the UC setting, computational implies statistical security for logarithmic protocols.

Theorem 30 (Computational Implies Statistical UC). *Let π and ρ be polynomial-time protocols with logarithmic communication complexity. Assume that π is as secure as ρ with respect to computational UC. Then π is as secure as ρ with respect to statistical UC.*

Proof. Analogous to the proof of Theorem 29, we can w.l.o.g. assume a logarithmic upper bound b on the communication complexity of environments Z, adversaries A, simulators S and the protocols π and ρ. We will implicitly assume this bound b for the rest of this proof.

In the case of statistical UC, we can assume that the environment Z just outputs its view, i.e., all messages it sent and received, since the distribution of Z's output can be (possibly inefficiently) computed from its communication.

As in the proof of Theorem 29, we construct sequences of oracle Turing machines $X_{A,k}$ and $Y_{S,k}$ and an efficiently computable logarithmic upper bound b' with the following properties:

- For all simulators S, adversaries A and environments Z that output its view, and for all $z \in \{0,1\}^*$, the distributions $X_{A,k}^{Z(1^k,z)}$ and $EXEC_{\pi,A,Z}(k,z)$ are identical, and so are the distributions $Y_{A,k}^{Z(1^k,z)}$ and $EXEC_{\pi,A,Z}(k,z)$.
- For all simulators S, adversaries A and environments Z, the oracle Turing machines $X_{A,k}^{Z(1^k,z)}$ and $Y_{A,k}^{Z(1^k,z)}$ call Z at most $b'(k)$ times and the total lenght of Z's answers is bounded by $b'(k)$.

Then let $\tilde{Z}$ be the environment/oracle that on security parameter k and auxiliary input z randomly chooses a sequence of $b'(k)$ messages of total length at most $b'(k)$. (The auxiliary input is ignored.) The environment $\tilde{Z}$ outputs its view.

By Lemma 27, if $\Delta(X_{A,k}^{Z(1^k,z)}; Y_{S,k}^{Z(1^k,z)})$ is not negligible in k, then $\Delta(X_{A,k}^{\tilde{Z}(1^k,z)}; Y_{S,k}^{\tilde{Z}(1^k,z)})$ is not negligible in k, either.

We now show that if π is as secure as ρ with respect to computational UC, then π is also as secure as ρ with respect to statistical UC. Let therefore an adversary A be given. By Lemma 28

we can w.l.o.g. assume A to be polynomial-time. Then, since π is as secure as ρ with respect to computational UC, there exists a polynomial-time simulator S such that $\{EXEC_{\pi,A,\tilde{Z}}(k,z_k)\}_k$ and $\{EXEC_{\rho,S,\tilde{Z}}(k,z_k)\}_k$ are computationally indistinguishable for any sequence z of strings. So $\{X_{A,k}^{\tilde{Z}(1^k,z_k)}\}_k$ and $\{Y_{A,k}^{\tilde{Z}(1^k,z_k)}\}_k$ are computationally indistinguishable, too.

Since $\tilde{Z}$ outputs its view by construction, we then have that $\{X_{A,k}^{\tilde{Z}(1^k,z_k)}\}_k$ and $\{Y_{A,k}^{\tilde{Z}(1^k,z_k)}\}_k$ are even statistically indistinguishable. Then for all environments Z that output only their view, also $\{X_{A,k}^{Z(1^k,z_k)}\}_k$ and $\{Y_{A,k}^{Z(1^k,z_k)}\}_k$ are statistically indistinguishable. Thus $\{EXEC_{\pi,A,Z}(k,z_k)\}_k$ and $\{EXEC_{\rho,S,Z}(k,z_k)\}_k$ are statistically indistinguishable. Since in the case of statistical UC it is sufficient to consider environments that output their view, it follows that π is as secure as ρ with respect to statistical UC. □

D Stand-Alone Security – Details and Proofs

We first give a definition of stand-alone security.

Definition 31 (Stand-Alone Security). *Let π and ρ be ITMs. We say that* π is as secure as ρ with respect to computational stand-alone security with auxiliary input, *if for every polynomial-time ITM A (the adversary) there is a polynomial-time ITM S (the simulator) such that for sequences x and z of strings of polynomial length, the families of distributions $\{\langle\!\langle A(1^k,z_k),\pi(1^k,x_k)\rangle\!\rangle\}_{k,z_k,x_k}$ and $\{\langle\!\langle S(1^k,z_k),\rho(1^k,x_k)\rangle\!\rangle\}_{k,z_k,x_k}$ are computationally indistinguishable in k.*

We speak of statistical stand-alone security with auxiliary input *if the above holds with unbounded A and S and statistical indistinguishability.*

We speak of computational/statistical stand-alone security without auxiliary input *if A and S do not get the additional input z_k (i.e. the distributions $\langle\!\langle A(1^k),\pi(1^k,x_k)\rangle\!\rangle$ and $\langle\!\langle S(1^k),\rho(1^k,x_k)\rangle\!\rangle$ are compared).*

Our definition is considerably simpler than that of e.g., [Gol04] since it abstracts away from details like the possibility of corruptions, asynchronous message delivery, and even the fact that there are different parties in the protocol.

However, it is easy to see that our results also hold for more complex definitions of stand-alone security since one can see our definition as a simple special case of a more general definition (the case of a single-party protocol), and the more general definition can be seen as a special case of our definition by including the corruption and network delivery mechanisms into the specification of the real or ideal protocol (i.e., the real protocol π in our model can be considered as being all protocol machines *and the network* in one machine, and similarly for the ideal protocol ρ).

In many models of stand-alone security, in the ideal model we do not allow arbitrary protocols, but only ideal functions. In order to be able to capture this restriction, we characterise the ITMs that correspond to such functions using the next definition.

Definition 32 (Function-Like ITMs). *We say an ITM is function-like if it sends only one message and receives only one message (in that order) and then gives output.*

D.1 On The Complexity of Finding a Good Adversary-Strategy

We are now going to construct protocols π and ρ such that there always exists a good (i.e., successful) adversary against the security of π (with respect to ρ), but such that finding that adversary is hard.

A central concept in our construction will be that of the adversary-polytope. The adversary-polytope of a protocol (with some fixed inputs) is the set of all distributions of the output of that protocols that can occur with various adversaries. If there is a distribution that can occur with some adversary in the real protocol but not in the ideal protocol (not even approximate), then π is not as secure as ρ with respect to statistical security. Reformulated in terms of adversary-polytopes, this condition reads as follows: There is a point in the adversary-polytope $\mathbf{A}_\pi$ of π that is (sufficiently far) outside the adversary-polytope $\mathbf{A}_\rho$ of ρ. However, if this point cannot be found efficiently, we still can hope for computational security. Thus our goal is to construct protocols π and ρ such that $\mathbf{A}_\pi$ contains a point sufficiently far outside of $\mathbf{A}_\rho$, but such that finding a point in $\mathbf{A}_\pi \setminus \mathbf{A}_\rho$ implies finding a witness to an NP-hard problem.

Definition 33 (Adversary-Polytope). *Let an ITM T be given. Assume that the output of T is in $\{1,\ldots,t\}$. For an ITM A, let $\langle A, T(x)\rangle$ denote the distribution of the output of T invoked with input x and running with A. Then we can consider $\langle A, T(x)\rangle$ as a vector $p \in \mathbb{R}^t$ by setting $p_i := \Pr[\langle A, T(x)\rangle = i]$. Then for some input x, the* adversary-polytope $\mathbf{A}_{T(x)} \subseteq \mathbb{R}^t$ *of $T(x)$ is defined as*

$$\mathbf{A}_{T(x)} := \{\langle A, T(x)\rangle : A \text{ is an ITM}\}.$$

To be able to speak more easily of adversaries that break the protocol, we give the following definition of good and strongly good adversaries.

Definition 34 (Good Adversaries). *Let π and ρ and A be ITMs, and let $\varepsilon > 0$.*

*We call A ε-*good for (π,ρ) *if for all ITMs S the statistical distance between $\langle\!\langle A, \pi\rangle\!\rangle$ and $\langle\!\langle S, \rho\rangle\!\rangle$ is bounded from above by ε. We call A good for (π,ρ) if A is ε-good for some $\varepsilon > 0$.*

We call A strongly ε-good for (π,ρ) *if for all ITMs S the statistical distance between $\langle A, \pi\rangle$ and $\langle S, \rho\rangle$ is bounded from above by ε. We call A strongly good for (π,ρ) if A is strongly ε-good for some $\varepsilon > 0$.*

It is easy to see that a protocol is statistically insecure iff there exist ε-good adversaries with sufficiently large ε. On the other hand, *strongly* ε-good adversaries exist iff there exists a point in A_π that is at least ε-far from A_ρ with respect to the 1-norm. So in general, the criterion in terms of adversary-polytopes does not necessarily coincide with the definition of stand-alone security. However, in all our constructions ε-good adversaries will be equivalent to strongly ε-good adversaries.

The NP-complete problem that we will reduce finding points in $\mathbf{A}_\pi \setminus \mathbf{A}_\rho$ to is the following:

Definition 35 (Set Cover). *Let $n \in \mathbb{N}$, $s_{ij} \in \{0,1\}$ with $i = 1,\ldots,m$ and $j = 1,\ldots,n$ and $d \leq n$. Let $s_{i\cdot} \neq 0$ for all $i = 1,\ldots,m$. Then (n,m,S,d) is an instance of set cover (with $S := ((s_{ij}))$). Let $S_j := \{i : s_{ij} = 1\}$. Then (n,m,S,d) is a yes-instance of set cover if there is a set $C \subseteq \{1,\ldots,n\}$ such that $\#C \leq d$ and $\bigcup_{j\in C} S_j = \{1,\ldots,m\}$.*

Note that set cover is NP-complete with a witness-preserving reduction, i.e., a SAT instance can be reduced to set cover such that a witness of the set cover instance can be transformed into a witness for the SAT instance in deterministic polynomial time. See e.g., [Pap93].

For the remainder of this section, we will often implicitly assume (n, m, S, d) to be a set cover instance and s_{ij} to be the components of S. So if in some lemma, definition, or proof an unqualified n, m, S, or d appears, it refers to the corresponding component of the set cover instance (n, m, S, d).

We now define a particular type of polytopes, the set cover polytopes. Such a set cover polytope encodes an instance of set cover and is additionally parametrised over an additional parameter $\varepsilon \in (0, 1)$. Here a high value of ε intuitively denotes that the set cover polytope encodes the set cover instance well. We will later see that if we can let the adversary-polytope $\mathbf{A}_\pi$ be a set cover polytope with sufficiently large ε and $\mathbf{A}_\rho$ a suitable cross-polytope, the points in $\mathbf{A}_\pi \setminus \mathbf{A}_\rho$ encode the witnesses for the encoded set cover instance.

Definition 36 (Set Cover Polytope). *Let (n, m, S, d) be an instance of set cover. Let $\varepsilon \in (0, 1)$. Let P be a polytope. We call P an* ε-set cover polytope for (n, m, S, d) *if the following holds:*

- $P \subseteq [0, 1]^n$.
- *Let $v \in \{0, 1\}^n$. If $\|v\|_1 \leq d$ and $\langle s_{i\cdot}, v\rangle \geq 1$ for all $i = 1, \ldots, m$, then $v \in P$.*
- *Let $v \in [0, 1]^n$. If $\|v\|_1 > d + 1 - \varepsilon$ or $\langle s_{i\cdot}, v\rangle < \varepsilon$ for some $i \in \{1, \ldots, m\}$, then $v \notin P$.*

An example of an ε-set cover polytope is the polytope given by the inequalities $v \in [0, 1]^n$, $\|v\|_1 \leq d$ and $\langle s_{i\cdot}, v\rangle \geq 1$. (It is an ε-set cover polytope for every $\varepsilon \in (0, 1)$).

The following lemma give us a reduction that maps points in a set cover polytope that have sufficiently large 1-norm (i.e., that are outside a suitable cross-polytope) to witnesses of set cover.

Lemma 37 (Reducing Set Cover to Polytope 1-Norm). *Let (n, m, S, d) be an instance of set cover, $\varepsilon \in (0, 1)$, $u := (1, \ldots, 1)^T \in \mathbb{R}^n$, and P an ε-set cover polytope. Let $P' := P - \frac{1}{2}u$.*

(i) If (n, m, S, d) is a yes-instance, then $\|P'\|_1 = \frac{n}{2}$.
(ii) If (n, m, S, d) is a no-instance, then $\|P'\|_1 \leq \frac{n}{2} - \frac{\varepsilon}{n^2+1}$.
(iii) Moreover, given a vector $v \in \mathbb{R}^n$ with $d_1(v, P') < \frac{\varepsilon}{n^2+1}$ and $\|v\|_1 > \frac{n}{2} - \frac{\varepsilon}{n^2+1}$, we can efficiently compute a witness for (n, m, S, d).

Proof. First, to show (i), assume that (n, m, S, d) is a yes-instance. Then there is a set C with $\#C \leq d$ such that $\bigcup_{j \in C} S_j = \{1, \ldots, m\}$. Let $v^* \in \mathbb{R}^n$ be defined by $v^*_j := 1$ if $j \in C$ and $v^*_j := 0$ otherwise. Since $\#C \leq d$ we have $\|v^*\|_1 \leq d$. Fix some $i \in \{1, \ldots, m\}$. Then $i \in \bigcup_{j \in C} S_j$, so we can choose some $j \in C$ with $s_{ij} = 1$. Since $s_{i\cdot} \geq 0$ and $v^* \geq 0$, we have $\langle s_{i\cdot}, v^*\rangle \geq s_{ij} v^*_j = 1$. Therefore $v^* \in P$. Let $v' := v^* - \frac{1}{2}u \in P'$. Since $v^* \in \{0, 1\}^n$, it is $|v'_j| = \frac{1}{2}$ for all j, thus $\|v'\|_1 = \frac{n}{2}$. So $\|P'\|_1 \geq \|v'\|_1 = \frac{n}{2}$. Since $P' \subseteq [-\frac{1}{2}, \frac{1}{2}]^n$, it is also $\|P'\|_1 \leq \|[-\frac{1}{2}, \frac{1}{2}]^n\|_1 = \frac{n}{2}$. Summarising, we get $\|P'\| = \frac{n}{2}$. This shows (i).

We proceed by showing (iii). Let $\delta := \frac{\varepsilon}{n^2+1}$ and assume that a vector $v \in \mathbb{R}^n$ is given with $d_1(v, P') < \delta$ and $\|v\|_1 > \frac{n}{2} - \delta$. We define v^* as follows: If $v_j \geq 0$, let $v^*_j := \frac{1}{2}$, otherwise

let $v_j^* := -\frac{1}{2}$. Since $d_1(v, P') < \delta$ and $P' \subseteq [-\frac{1}{2}, \frac{1}{2}]^n$, it is $d_1(v, [-\frac{1}{2}, \frac{1}{2}]^n) < \delta$ and therefore $|v_j| < \frac{1}{2} + \delta$ for all j. Furthermore, since $\|v\|_1 > \frac{n}{2} - \delta$, it is $\sum_j |v_j| > \frac{n}{2} - \delta$. It follows that $|v_j| > \frac{1}{2} - n\delta$ for all j. Then $|v_j - v_j^*| < n\delta$ for all j and thus $d_1(v, v^*) < n^2\delta$.

Since $d_1(v, P') < \delta$, there is a $v' \in P'$ with $d_1(v, v') < \delta$ (the existence is sufficient, we do not need to compute v'). Then $\|v'\|_1 > \|v\|_1 - \delta > \frac{n}{2} - 2\delta$. Furthermore, $d_1(v', v^*) \leq d_1(v, v') + d_1(v, v^*) < (n^2 + 1)\delta$. Let $\tilde{v}' := v' + \frac{1}{2}u$ and $\tilde{v}^* := v^* + \frac{1}{2}u$. Then $\tilde{v}' \in P$ and $\tilde{v}^* \in \{0, 1\}^n$. Since P is an ε-set cover polytope, it is $\|\tilde{v}'\|_1 \leq d + 1 - \varepsilon$ and $\langle s_{i\cdot}, \tilde{v}' \rangle \geq \varepsilon$ for all $i \in \{1, \ldots, m\}$. Thus we have $\|\tilde{v}^*\|_1 \leq \|\tilde{v}'\|_1 + d_1(v', v^*) < d + 1 - \varepsilon + (n^2 + 1)\delta = d + 1$. Since $\tilde{v}^* \in \{0, 1\}^n$, the value $\|\tilde{v}^*\|_1$ is an integer. Therefore $\|\tilde{v}^*\| \leq d$. Since all $s_{ij} \in \{0, 1\}$, we have further $\langle s_{i\cdot}, \tilde{v}^* \rangle \geq \langle s_{i\cdot}, \tilde{v}' \rangle - d_1(v', v^*) > \varepsilon - (n^2 + 1)\delta = 0$. Since $\langle s_{i\cdot}, \tilde{v}^* \rangle$ is an integer, it follows that $\langle s_{i\cdot}, \tilde{v}^* \rangle \geq 1$ for all i. Let $C := \{j : \tilde{v}_j^* = 1\} \subseteq \{1, \ldots, n\}$. We will show that C is a witness for the set cover instance (n, m, S, d). Since $\tilde{v}^* \in \{0, 1\}^*$ and $\|v^*\|_1 \leq d$, we have $\#C \leq d$. Fix some $i \in \{1, \ldots, m\}$. Then $\langle s_{i\cdot}, \tilde{v}^* \rangle \geq 1$ and since $s_{ij}, v_j \in \{0, 1\}$ for all j, there is a $j \in \{1, \ldots, n\}$ such that $v_j = 1$ and $s_{ij} = 1$. Since $v_j = 1$, we have $j \in C$, and since $s_{ij} = 1$, we have $i \in S_j \subseteq \bigcup_{j \in C} S_j$. Since this holds for all $i \in \{1, \ldots, m\}$, the set C is a witness for (n, m, S, d). Since $j \in C$ if and only if $v_j \geq 0$, we can efficiently compute C from v. This proves (iii).

To show (ii), it is sufficient to note that if $\|P'\| > \frac{n}{2} - \frac{\varepsilon}{n^2+1}$, there exists a $v \in P'$ such that $\|v\|_1 = \|P'\|_1 > \frac{n}{2} - \frac{\varepsilon}{n^2+1}$ (note that P' is a polytope and thus closed). Since $v \in P'$, it is $d_1(v, P') = 0 < \frac{\varepsilon}{n^2+1}$. So by (iii) there exists a witness for (n, m, S, d), in contradiction to the assumption that (n, m, S, d) is a no-instance. □

We are now going to construct a set cover polytope for a given instance of set cover from simple polytopes. This construction we will later transform into a recursive definition of a protocol π whose adversary-polytope will then also be a set cover polytope.

The following definition states the building blocks of our recursive construction:

Definition 38. *For $l \in [0, 1]$ and $\varepsilon \in (0, 1)$ and $i \in \{1, \ldots, m\}$ and $x, y \in \{0, 1\}^n$ and $g \in \{0, \ldots, \|x \cdot y\|_1\}$, we define the following sets:*

- *The* upper bound polytope $D_l := \operatorname{conv}\{v \in \{0, l\}^n : \|v\|_1 \leq ld\}$
- *The* lower bound polytope $S_l^i := \operatorname{conv}\{v \in \{0, l\}^n : \langle s_{i\cdot}, v \rangle \geq l\}$
- *The* combined polytope $P_\varepsilon := D_{1-m\varepsilon} + \sum_{i=1}^m S_\varepsilon^i$
- *The* recursive vertex set $V_g^{x,y} := \{v \in \{0, 1\}^n : \langle x \cdot y, v \rangle \leq g,\ v \leq y\}$
- *The* recursive polytope $C_g^{x,y} := \operatorname{conv} V_g^{x,y}$.

Note that these sets implicitly depend on the set cover instance (n, m, S, d).

The next lemma will show that P_ε is indeed an ε-set cover polytope, and the Lemmas 40–43 thereafter will allow to recursively construct P_ε from polytopes of form $C_g^{x,y}$ with $g \in \{0, 1\}$. These polytopes $C_g^{x,y}$ have at most two vertices and are therefore very easy to construct.

Lemma 39. *If $0 < \varepsilon \leq \frac{1}{nm+1}$ then P_ε is an ε-set cover polytope.*

Proof. Since $D_{1-m\varepsilon} \subseteq [0, 1-m\varepsilon]^n$ and $S^i_\varepsilon \subseteq [0,\varepsilon]^n$, it is $P_\varepsilon = D_{1-m\varepsilon} + \sum_{i=1}^m S^i_\varepsilon \subseteq [0,1-m\varepsilon]^n + m \cdot [0,\varepsilon]^n = [0,1]^n$.

Let a vector $v \in \{0,1\}^n$ be given with $\|v\|_1 \leq d$ and $\langle s_{i\cdot}, v\rangle \geq 1$ for all i. Then $(1-m\varepsilon)v \in \{0, 1-m\varepsilon\}^n$ and $\|(1-m\varepsilon)v\|_1 \leq (1-m\varepsilon)d$, so $(1-m\varepsilon)v \in D_{1-m\varepsilon}$. Further, $\varepsilon v \in \{0,\varepsilon\}^n$ and $\langle s_{i\cdot}, \varepsilon v\rangle \geq \varepsilon$, so $\varepsilon v \in S^i_\varepsilon$ for all i. Thus $v = (1-m\varepsilon)v + \sum_{i=1}^m \varepsilon v \in D_{1-m\varepsilon} + \sum_{i=1}^m S^i_\varepsilon = P_\varepsilon$.

Let now a vector $v \in [0,1]^n$ be given with $\|v\|_1 > d + 1 - \varepsilon$. Since $\|P_\varepsilon\|_1 \leq \|D_{1-m\varepsilon}\|_1 + \sum_{i=1}^m \|S^i_\varepsilon\|_1 \leq (1-m\varepsilon)d + \sum_{i=1}^m n\varepsilon \leq d + 1 - \varepsilon$, it follows $v \notin P_\varepsilon$.

Now, let a vector $v \in [0,1]^n$ be given with $\langle s_{i\cdot}, v\rangle < \varepsilon$ for some i. Assume that $v \in P_\varepsilon$. Then we can write $v = v_a + v_b$ with $v_a \in S^i_\varepsilon$ and $v_b \in D_{1-m\varepsilon} + \sum_{k\neq i} S^k_\varepsilon$. Then $v_b \geq 0$, so $\langle s_{i\cdot}, v_b\rangle \geq 0$. Since $v_a \in S^i_\varepsilon$, it is $\langle s_{i\cdot}, v_a\rangle \geq \varepsilon$. Therefore we have $\langle s_{i\cdot}, v\rangle = \langle s_{i\cdot}, v_a\rangle + \langle s_{i\cdot}, v_b\rangle \geq \varepsilon$. This is a contradiction, hence $v \notin P_\varepsilon$.

We have verified all properties given in Definition 36, hence P_ε is an ε-set cover polytope. □

The following lemma allows to write P_ε in terms of polytopes $C^{x,y}_g$ (but we do not necessarily have $g \in \{0,1\}$, so these polytopes $C^{x,y}_g$ may still be complex).

Lemma 40. *Let $u := (1,\dots,1) \in \{0,1\}^n$ and $l \in [0,1]$. Then $D_l = l \cdot C^{u,u}_d$ and $S^i_l = l \cdot (u - C^{s_{i\cdot},u}_{\|s_{i\cdot}\|_1 - 1})$.*

Proof. Since for any $v \in \{0,1\}^n$ we have $\|v\|_1 = \langle u, v\rangle$ and $v \leq u$, we have $D_1 = C^{u,u}_d$ and thus $D_l = l \cdot D_1 = l \cdot C^{u,u}_d$. Since $s_{i\cdot} \geq 0$ we have $\langle s_{i\cdot}, u - v\rangle = \|s_{i\cdot}\|_1 - \langle s_{i\cdot}, v\rangle$ and thus

$$\begin{aligned} S^i_1 &= \operatorname{conv}\{v \in \{0,1\}^n : \langle s_{i\cdot}, v\rangle \geq 1\} \\ &= \operatorname{conv}\{v \in \{0,1\}^n : \langle s_{i\cdot}, u - v\rangle \leq \|s_{i\cdot}\|_1 - 1\} \\ &= \operatorname{conv}\{u - v' : v' \in \{0,1\}^n,\ \langle s_{i\cdot}, v'\rangle \leq \|s_{i\cdot}\|_1 - 1\} \\ &= \operatorname{conv}(u - V^{s_{i\cdot},u}_{\|s_{i\cdot}\|_1 - 1}) = u - C^{s_{i\cdot},u}_{\|s_{i\cdot}\|_1 - 1}. \end{aligned}$$

From this it follows that $S^i_l = l \cdot S^i_1 = l \cdot (u - C^{s_{i\cdot},u}_{\|s_{i\cdot}\|_1 - 1})$. □

The following lemma states some probably well-known facts about the convex hulls of sums and unions of sets. We give the proof for completeness.

Lemma 41. *For sets $A, B, A_1, \dots, A_r \subseteq \mathbb{R}^n$, it is $\operatorname{conv}\bigcup_i A_i = \operatorname{conv}\bigcup_i \operatorname{conv} A_i$ and $\operatorname{conv}(A+B) = \operatorname{conv} A + \operatorname{conv} B$.*

Proof. Since $A_i \subseteq \operatorname{conv} A_i$ we have $\operatorname{conv}\bigcup_i A_i \subseteq \operatorname{conv}\bigcup_i \operatorname{conv} A_i$. To show the other direction, we first show $\bigcup_i \operatorname{conv} A_i \subseteq \operatorname{conv}\bigcup_i A_i$. Let $x \in \bigcup_i \operatorname{conv} A_i$ be given. Then there is an i such that $x = \sum_j r_j x_j$ with $\sum_j r_j = 1$, $r_j \geq 0$ and $x_j \in A_i \subseteq \bigcup_i A_i$. Thus $x \in \operatorname{conv}\bigcup_i A_i$ and therefore $\bigcup_i \operatorname{conv} A_i \subseteq \operatorname{conv}\bigcup_i A_i$. From this it follows that $\operatorname{conv}\bigcup_i \operatorname{conv} A_i \subseteq \operatorname{conv}\operatorname{conv}\bigcup_i A_i = \operatorname{conv}\bigcup_i A_i$. So we have shown $\operatorname{conv}\bigcup_i A_i = \operatorname{conv}\bigcup_i \operatorname{conv} A_i$.

We now show $\operatorname{conv}(A+B) \subseteq \operatorname{conv} A + \operatorname{conv} B$. Let $x \in \operatorname{conv}(A+B)$ be given. Then $x = \sum_j r_j(a_j + b_j)$ with $\sum_j r_j = 1$ and $r_j \geq 0$. With $x_a := \sum_j r_j a_j$ and $x_b := \sum_j r_j b_j$ we have $x = x_a + x_b \in \operatorname{conv} A + \operatorname{conv} B$.

Now, we show $\operatorname{conv} A + \operatorname{conv} B \subseteq \operatorname{conv}(A+B)$. Let $x \in \operatorname{conv} A + \operatorname{conv} B$ be given. Then we can write $x = \sum_j r_j c_j + \sum_j s_j a_j + \sum_j t_j b_j$ with $c_j \in A+B$ and $a_j \in A$ and $b_j \in B$ and $\sum_j r_j + \sum_j s_j = 1$ and $\sum_j r_j + \sum_j t_j = 1$ and $r_j, s_j, t_j \geq 0$. We call the number of summands of the form $s_j a_j$ and $t_j b_j$ (but not $r_j c_j$) the degree of the representation. Let $x = \sum_j r_j c_j + \sum_j s_j a_j + \sum_j t_j b_j$ be such a representation with minimal degree. Then all $s_j, t_j > 0$. Assume that minimal degree is greater than 0. Then the second and the third sum both contain at least one summand, since otherwise $\sum_j s_j = \sum_j t_j = 1 - \sum_j r_j$. Then there exists an s_k or a t_k that is minimal under all s_j and t_j. W.l.o.g., we assume that some s_k is minimal. Then $s_k \leq t_1$. Let $t'_1 := t_1 - s_k$ and $t'_j := t'_j$ for $j > 1$. Let $c_0 := a_k + b_1$ and $r_0 := s_k$. We then have

$$x = \sum_j r_j c_j + \sum_j s_j a_j + \sum_j t_j b_j = r_0 c_0 + \sum_j r_j c_j + \sum_{j \neq k} s_j a_j + \sum_j t'_j b_j.$$

Since the right hand side is a representation of x with smaller degree this is a contradiction to the minimality of the original representation. So the original representation had degree 0, so $x = \sum_j r_j c_j$ with $\sum_j r_j = 1$ and $r_j \geq 0$ and $c_j \in A+B$. Thus $x \in \operatorname{conv}(A+B)$.

It follows that $\operatorname{conv} A + \operatorname{conv} B = \operatorname{conv}(A+B)$. □

The following definition provides some notation to write Lemma 43 more concisely.

Definition 42. *For a vector $y \in \{0,1\}^n$ with $w := \|y\|_1$, let $l(y)$ and $r(y)$ be the vectors with the first $\lceil \frac{w}{2} \rceil$ and last $\lfloor \frac{w}{2} \rfloor$ bits of y set, respectively.*

Formally, let $l(y)$ and $r(y)$ be the unique vectors satisfying

$$l(y), r(y) \in \{0,1\}^n \quad \textit{and} \quad y = l(y) + r(y) \quad \textit{and} \quad \|l(y)\| = \lceil \tfrac{w}{2} \rceil \quad \textit{and}$$
$$\|r(y)\| = \lfloor \tfrac{w}{2} \rfloor \quad \textit{and} \quad l(y)_i = r(y)_j = 1 \Rightarrow i < j.$$

The following lemma gives a recursive construction of $C_g^{x,y}$ using sums and unions. The base case of the recursion are sets $C_g^{x,y}$ with $g \in \{0,1\}$ and the depth of the recursion is logarithmic in $\|x \cdot y\|_1$ and its width is polynomial in $\|x \cdot y\|_1$. Together with the preceding lemmas, we can now recursively construct P_ε using only sums and unions of convex sets.

Lemma 43. *For $x, y \in \{0,1\}^n$ and $0 \leq g \leq \|x \cdot y\|_1$, we have*

$$C_g^{x,y} = \operatorname{conv}\left(\bigcup_I \left(C_i^{x,l(y)} + C_{g-i}^{x,r(y)} \right) \right)$$

where $I := \{i = 0, \ldots, g : i \leq \|x \cdot l(y)\|_1$ and $g - i \leq \|x \cdot r(y)\|_1\}$.

Proof. Let $c_l := \|x \cdot l(y)\|_1$ and $c_r := \|x \cdot r(y)\|_1$. We first show $V_g^{x,y} = \bigcup_I \left(V_i^{x,l(y)} + V_{g-i}^{x,r(y)} \right)$ where $I := \{i = 0, \ldots, g : i \leq c_l$ and $g - i \leq c_r\}$.

Let $v \in V_g^{x,y}$. Then by definition, $v \in \{0,1\}^n$ and $\langle x \cdot y, v \rangle \leq g$ and $v \leq y$. Let $v_l := v \cdot l(y) \leq l(y)$ and $v_r := v \cdot r(y) \leq r(y)$. Let $a := \langle x \cdot y, v_l \rangle = \langle x \cdot l(y), v_l \rangle \leq c_l$. Then $v_l \in V_a^{x,l(y)}$.

Similarly, let $b := \langle x \cdot y, v_r \rangle = \langle x \cdot r(y), v_r \rangle \leq c_r$. Then $v_r \in V_b^{x,r(y)}$.

Since $l(y) + r(y) = y$, we have $v = v \cdot y = v_l + v_r$ and therefore $a + b \leq \langle x \cdot y, v \rangle \leq g$. Furthermore, $g \leq \|x \cdot y\|_1 = c_l + c_r$. So we have $a + b \leq g \leq c_l + c_r$ and $a \leq c_l$ and $b \leq c_r$. Therefore it exists an integer $0 \leq i \leq g$ with $a \leq i \leq c_l$ and $b \leq g - i \leq c_r$. Then $v_l \in V_b^{x,l(y)} \subseteq V_i^{x,l(y)}$ and $v_r \in V_b^{x,r(y)} \subseteq V_{g-i}^{x,r(y)}$. Thus $v = v_l + v_r \in \bigcup_I \Big(V_i^{x,l(y)} + V_{g-i}^{x,r(y)} \Big)$.

Therefore we have $V_g^{x,y} \subseteq \bigcup_I \Big(V_i^{x,l(y)} + V_{g-i}^{x,r(y)} \Big)$.

Let now some $v \in \bigcup_I \Big(V_i^{x,l(y)} + V_{g-i}^{x,r(y)} \Big)$ be given. Then there exist an $i \in I$ and vectors $v_l \in V_i^{x,l(y)}$ and $v_r \in V_{g-i}^{x,r(y)}$ such that $v = v_l + v_r$. Since $v_l \leq l(y)$, we have $\langle x \cdot y, v_l \rangle = \langle x \cdot l(y), v_l \rangle$. Analogously, it follows $\langle x \cdot y, v_r \rangle = \langle x \cdot r(y), v_r \rangle$. Hence $\langle x \cdot y, v \rangle = \langle x \cdot y, v_l \rangle + \langle x \cdot y, v_r \rangle = \langle x \cdot l(y), v_l \rangle + \langle x \cdot r(y), v_r \rangle \leq i + g - i = g$. Furthermore, $v = v_l + v_r \leq l(y) + r(y) = y$. And since $v_l, v_r \in \{0,1\}^n$ and $v \leq y \in \{0,1\}^n$, it is $v \in \{0,1\}^n$. Hence $v \in V_g^{x,y}$.

Therefore we have $V_g^{x,y} \subseteq \bigcup_I \Big(V_i^{x,l(y)} + V_{g-i}^{x,r(y)} \Big)$.

Applying Lemma 41 twice (marked with $(*)$), it follows

$$\begin{aligned} C_g^{x,y} = \operatorname{conv} V_g^{x,y} &= \operatorname{conv}\Big(\bigcup_I \big(V_i^{x,l(y)} + V_{g-i}^{x,r(y)} \big) \Big) \\ &\overset{(*)}{=} \operatorname{conv}\Big(\bigcup_I \operatorname{conv}\big(V_i^{x,l(y)} + V_{g-i}^{x,r(y)} \big) \Big) \\ &\overset{(*)}{=} \operatorname{conv}\Big(\bigcup_I \big(\operatorname{conv} V_i^{x,l(y)} + \operatorname{conv} V_{g-i}^{x,r(y)} \big) \Big) \\ &= \operatorname{conv}\Big(\bigcup_I \big(C_i^{x,l(y)} + C_{g-i}^{x,r(y)} \big) \Big). \end{aligned}$$

□

We now have a recursive construction of a set cover polytope using sums and unions of sets. In the following, we will transform this construction into a recursive construction of a protocol π whose adversary-polytope is a set cover polytope (up to affine transformation).

In order to do so, we first need to be able to express sums and unions of adversary-polytopes by operations on protocols. The following lemma gives us the means to do so.

Lemma 44 (Constructions of Adversary-Polytopes). *Let T an ITM. Let $r_1, \ldots, r_q \geq 0$ with $\sum_i r_i = 1$. Let $x, x_1, \ldots, x_q \in \Sigma^*$. Let R be an ITM that upon input x chooses some value i with probability r_i, sends i, and then executes $T(x_i)$. Then $\mathbf{A}_{R(x)} = \sum r_i \mathbf{A}_{T(x_i)}$.*

Let U be an ITM that upon input x expects a message $i \in \{1, \ldots, q\}$ and then executes $T(x_i)$. (If U receives a message of different form, it assumes $i = 1$.) Then $\mathbf{A}_{U(x)} = \operatorname{conv} \bigcup_i \mathbf{A}_{T(x_i)}$.

Proof. First, we show that $\mathbf{A}_{R(x)} = \sum_i r_i \mathbf{A}_{T(x_i)}$.

Let $v \in \mathbf{A}_{R(x)}$. Then there exists an ITM A such that $\langle A, R(x) \rangle = v$. Let R_i be the ITM that behaves as $R(x)$, except that it always chooses message i. Then $\langle A, R(x) \rangle = \sum_i r_i \langle A, R_i \rangle$. Since R_i behaves like $T(x_i)$, except that it first sends a fixed message to A, we can construct an ITM A' from A that does not expect this first message but assumes it to be i, and get $\langle A, R_i \rangle = \langle A', T(x_i) \rangle \in \mathbf{A}_{T(x_i)}$. Thus $v \in \sum_i r_i \mathbf{A}_{T(x_i)}$. So $\mathbf{A}_{R(x)} \subseteq \sum_i r_i \mathbf{A}_{T(x_i)}$.

Now, let $v \in \sum_i r_i \mathbf{A}_{T(x_i)}$. Then $v = \sum_i r_i v^{(i)}$ with $v^{(i)} \in \mathbf{A}_{T(x_i)}$ and there are ITMs A_i such that $v^{(i)} = \langle A_i, T(x_i)\rangle$ for all i. Let A be the ITM that expects a message i and then executes A_i. Since each i is chosen with probability r_i by $R(x)$, we have $\langle A, R(x)\rangle = \sum_i r_i \langle A_i, T(x_i)\rangle = v$, so $v \in \mathbf{A}_{R(x)}$. Thus $\mathbf{A}_{R(x)} = \sum_i r_i \mathbf{A}_{T(x_i)}$.

We proceed by showing $\mathbf{A}_{U(x)} = \operatorname{conv} \bigcup_i \mathbf{A}_{T(x_i)}$.

Let $v \in \mathbf{A}_{U(x)}$. Then there is an ITM A such that $v = \langle A, U(x)\rangle$. W.l.o.g., we assume that the first message sent by A is in $\{1, \dots, q\}$. Let r_i be the probability that that first message is i. Then $\langle A, U(x)\rangle = \sum_i r_i \langle A_i, T(x_i)\rangle$ where A_i is the residual ITM A after sending i. Then $v^{(i)} := \langle A_i, T(x_i)\rangle \in \mathrm{A}_{T(x_i)} \subseteq \bigcup_i \mathrm{A}_{T(x_i)}$, so $v = \sum_i r_i v^{(i)} \in \operatorname{conv} \bigcup_i \mathrm{A}_{T(x_i)}$. It follows that $\mathbf{A}_{U(x)} \subseteq \operatorname{conv} \bigcup_i \mathrm{A}_{T(x_i)}$.

Now, let $v \in \operatorname{conv} \bigcup_i \mathrm{A}_{T(x_i)}$. Then we can decompose v such that $v = \sum_i r_i v^{(i)}$ with $\sum_i r_i = 1$, $r_i \geq 0$ and $v^{(i)} \in \operatorname{conv} \mathrm{A}_{T(x_i)} = \mathrm{A}_{T(x_i)}$. So there are ITMs A_i such that $v^{(i)} = \langle A_i, T(x_i)\rangle$ for all i. Let A be the ITM that chooses an $i \in \{1, \dots, q\}$ with probability r_i and then executes A_i. Then $\langle A, U(x)\rangle = \sum r_i \langle A_i, T(x_i)\rangle = v$. So $v \in \mathrm{A}_{U(x)}$. It follows that $\mathbf{A}_{U(x)} = \operatorname{conv} \bigcup_i \mathbf{A}_{T(x_i)}$. □

The reader may have noticed that Lemma 44 does not give us the possibility to construct an adversary-polytope $\mathbf{A}_R$ that is the sum $\mathbf{A}_{T(1)} + \mathbf{A}_{T(2)}$ of two adversary-polytopes $\mathbf{A}_{T(1)}$ and $\mathbf{A}_{T(2)}$, but only a downscaled sum $\frac{1}{2}(\mathbf{A}_{T(1)} + \mathbf{A}_{T(2)})$. This is to be expected since the sum of two sets of probability distributions (considered as points in $\mathbb{R}^n$) is not necessarily a set of probability distributions. Therefore, we cannot directly map the recursive construction from Lemma 43 into a recursive construction of a protocol π. Instead, we have to keep track of the additional downscaling of the polytopes. Similarly, we will also have to transform the occurring polytopes so that they will be a subset of the set of all probability distributions.

To keep track of these scalings and translations we will use affine transformations that map the unit cube in $\mathbb{R}^n$ into the set of probability distributions considered as a subset of $\mathbb{R}^{n+1}$. Such maps we call valid.

Definition 45 (Valid Affine Maps). *An affine map $f : \mathbb{R}^n \to \mathbb{R}^{n+1}$ is called* valid *if $f([0,1]^n) \subseteq \{x \in \mathbb{R}^{n+1} : x \geq 0, \|x\|_1 = 1\}$ and if $f(x) = Ax + b$ for some rational matrix A and rational vector b.*

We can now start to construct our protocol. First, we construct an ITM H that given a set of points as input constructs a polytope with that vertices (up to transformation by a valid affine map). Given this ITM we can then construct adversary-polytopes that have few vertices.

Lemma 46. *There exists a polynomial-time ITM H such that for $n \in \mathbb{N}$, a finite nonempty set $X \subseteq [0,1]^n \cap \mathbb{Q}^n$, and a valid affine map f, we have $\mathbf{A}_{H(n,X,f)} = f(\operatorname{conv} X)$. Upon input (n, X, f), the ITM H has communication complexity $\lceil \log \# X \rceil$. The ITM H is function-like.*

Proof. Let V be the ITM that upon input $v \in \mathbb{Q}^{n+1}$ with $\sum v_i = 1$ and $v_i \geq 0$ for all i chooses a value $i \in \{1, \dots, n+1\}$ with probability v_i. Then V sends i to the other ITM and outputs i. Then $\langle A, V(v)\rangle = v$ for all valid inputs v and all ITMs A, so $\mathbf{A}_{V(v)} = \{v\}$.

The ITM H behaves as follows: Upon input (n, X, f) satisfying the conditions given in the statement of this lemma, it enumerates X (in some deterministic fashion) such that

$\{x_1, \ldots, x_{\#X}\}$. Then it expects a message $i \in \{1, \ldots, x_{\#X}\}$ and runs $V(f(x_i))$. Since f is a valid affine map, $f(x_i)$ is a valid input for V. So by Lemma 44 we have

$$\mathbf{A}_{H(n,X,f)} = \operatorname{conv} \bigcup_i \mathbf{A}_{V(f(x_i))} = \operatorname{conv}\{f(x_1), \ldots, f(x_{\#X})\} = f(\operatorname{conv} X).$$

Since an $i \in \{1, \ldots, \#X\}$ can be encoded in $\lceil \log \#X \rceil$ bits, H has communication complexity $\lceil \log \#X \rceil$. Obviously, H is function-like and runs in polynomial time. □

We can now transform the recursion for $C_g^{x,y}$ given by Lemma 43 into a construction of a protocol that has $C_g^{x,y}$ as an adversary-polytope (up to an affine transformation). For the base case of the recursion we use the ITM H from Lemma 46 and the sums and unions are handled using the constructions from Lemma 44. The resulting adversary-polytope is a downscaled version of $C_g^{x,y}$ since we cannot directly construct sums, however, it will only be downscaled by a polynomial factor which turns out to be good enough for our purposes.

Lemma 47. *There exists a polynomial-time ITM C such that for all $n \in \mathbb{N}$, $x \in \{0,1\}^n$, $y \in \{0,1\}^n \setminus \{0\}^n$, $g \in \{0, \ldots, \|x \cdot y\|_1\}$ and all valid affine maps $f : \mathbb{R}^n \to \mathbb{R}^{n+1}$, we have*

- *The communication complexity of C upon input (n, x, y, g, f) is $O\big((\log n)^2\big)$.*
- *The adversary-polytope of $C(n, x, y, g, f)$ is*

$$\mathbf{A}_{C(n,x,y,g,f)} = f\big(\lambda(\|y\|_1) \cdot C_g^{x,y}\big).$$

where $\lambda(y) := 2^{-\lceil \log \|y\|_1 \rceil}$.

Proof. We call an input tuple (n, x, y, g, f) valid if it satisfies the conditions given in the lemma, i.e., if $n \in \mathbb{N}$, $x \in \{0,1\}^n$, $y \in \{0,1\}^n \setminus \{0\}^n$, $g \in \{0, \ldots, \|x \cdot y\|_1\}$ and f is a valid affine map.

For $\|y\|_1 \geq 2$, define h as follows: If $\left\lceil \log \left\lceil \frac{\|y\|_1}{2} \right\rceil \right\rceil = \left\lceil \log \left\lfloor \frac{\|y\|_1}{2} \right\rfloor \right\rceil$, let $h := 1$. Otherwise, let $h := \frac{1}{2}$. If f is a valid affine map, so is $f'(v) := f(hv)$.

We define C recursively. Upon valid input (n, x, y, g, f) with $\|y\|_1 \geq 2$, it behaves as follows:

- Let I be defined as in Lemma 43. Note that I is nonempty.
- First, C expects an $i \in I$ from the adversary. If no $i \in I$ is received, C sets $i := \min I$ otherwise.
- Then, C chooses a uniformly random bit $b \in \{0,1\}$ and sends b to the adversary.
- If $b = 0$, the ITM C executes $C(n, x, l(y), i, f)$ and otherwise $C(n, x, r(y), g - i, f')$. (For the functions r and l see Definition 42.)

Upon valid input (n, x, y, g, f) with $\|y\|_1 = 1$, we compute $V := V_g^{x,y}$ and execute the ITM $H(n, V, f)$ from Lemma 46. (Note that in this case $\#V_g^{x,y} \leq 2$.)

Upon invalid input, C terminates with output 1.

For the recursion to make sense, we first have to verify that the ITMs C that are invoked as subprograms are always invoked with valid input. In this case of H this is straight-forward: $V_g^{x,y} \subseteq \{0,1\}^n$ and f is a valid map.

To see that C is invoked with valid input, first consider the case $b = 0$. In this case, we have to verify that $n \in \mathbb{N}$, $x \in \{0,1\}^n$, $l(y) \in \{0,1\}^n$, $i \in \{0, \dots, \|x \cdot y\|_1\}$ and f is a valid affine map. The conditions for n, x and f are satisfied by assumption. Since $\|y\|_1 \geq 2$, we have $l(y) \in \{0,1\}^n \setminus \{0\}^n$ by definition of $l(y)$ and we have $i \in \{0, \dots, \|x \cdot l(y)\|_1\}$ by definition of I. In the case $b = 1$ the map f' is valid as seen above and we have $r(y) \in \{0,1\}^n \setminus \{0\}^n$ by definition of $r(y)$ and $g - i \in \{0, \dots, \|x \cdot r(y)\|_1\}$ by definition of I.

Note that the recursion terminates since for $\|y\|_1 \geq 2$, $\|l(y)\|_1, \|r(y)\|_1 < \|y\|_1$. Moreover, since $\|l(y)\|_1, \|r(y)\|_1 \leq \lceil \|y\|_1/2 \rceil$, the recursion has at most logarithmic depth (and no branching takes place), so in particular, C is polynomial-time.

We now examine the communication complexity of C. In each round (with exception of the last, where H is invoked), C receives an element from $I \subseteq \{1, \dots, n\}$ and sends a bit b. Therefore the communication complexity within one round is $O(\log n)$. In the last round H is invoked. Since $\# V_g^{x,y} \leq 2$ in this case, the communication complexity of H is $O(1)$.

Since there are $O(\log n)$ rounds, the overall communication complexity is $O\big((\log n)^2\big)$.

It is left to show that $\mathbf{A}_{C(n,x,y,g,f)}$ has the required form.

For $\|y\|_1 = 1$ it is $\lambda(y) = 1$. By construction of H, we then have

$$\mathbf{A}_{C(n,x,y,g,f)} = \mathbf{A}_{H(n,V_g^{x,y},f)} = f\big(\mathrm{conv}(V_g^{x,y})\big) = f(\lambda(y) \cdot C_g^{x,y}).$$

Now consider the case $\|y\|_1 \geq 2$. It is $\lceil \log \|y\|_1 \rceil = \left\lceil \log \left\lceil \frac{\|y\|_1}{2} \right\rceil \right\rceil + 1 = \lceil \log \|l(y)\|_1 \rceil + 1$. Thus $\lambda(y) = \frac{1}{2}\lambda(l(y))$. Further, by definition of h, we have

$$h 2^{-\lceil \log \|r(y)\|_1 \rceil} = h 2^{-\left\lceil \log \left\lfloor \frac{\|y\|_1}{2} \right\rfloor \right\rceil} = 2^{-\left\lceil \log \left\lceil \frac{\|y\|_1}{2} \right\rceil \right\rceil} = 2^{-\lceil \log \|l(y)\|_1 \rceil} = \lambda(l(y)).$$

So summarising, we have $\frac{1}{2}\lambda(l(y)) = \frac{1}{2} h \lambda(r(y)) = \lambda(y)$.

Then, by induction we get

$$\begin{aligned}
\mathbf{A}_{C(n,x,y,g,f)} &\overset{(*)}{=} \mathrm{conv} \bigcup_{i \in I} \big(\tfrac{1}{2} \mathbf{A}_{C(n,x,l(y),i,f)} + \tfrac{1}{2} \mathbf{A}_{C(n,x,r(y),g-i,f')} \big) \\
&\overset{(**)}{=} \mathrm{conv} \bigcup_{i \in I} \Big(\tfrac{1}{2} f\big(\lambda(l(y)) \cdot C_i^{x,l(y)}\big) + \tfrac{1}{2} f\big(h\lambda(r(y)) \cdot C_{g-i}^{x,r(y)}\big) \Big) \\
&= \mathrm{conv} \bigcup_{i \in I} \Big(f\big(\tfrac{1}{2}\lambda(l(y)) \cdot C_i^{x,l(y)} + \tfrac{1}{2} h\lambda(r(y)) \cdot C_{g-i}^{x,r(y)} \big) \Big) \\
&= \mathrm{conv} \bigcup_{i \in I} \Big(f\big(\lambda(y) \cdot \big(C_i^{x,l(y)} + C_{g-i}^{x,r(y)} \big) \big) \Big) \\
&= f\Big(\lambda(y) \cdot \mathrm{conv} \bigcup_{i \in I} \big(C_i^{x,l(y)} + C_{g-i}^{x,r(y)} \big) \Big) \\
&\overset{(***)}{=} f\big(\lambda(y) \cdot C_g^{x,y}\big).
\end{aligned}$$

We used Lemma 44 for $(*)$, the induction hypothesis for $(**)$, and Lemma 43 for $(***)$.

So for all valid inputs, we have $\mathbf{A}_{C(n,x,y,g,f)} = f\big(\lambda(y) \cdot C_g^{x,y}\big)$. □

Finally, since we can construct P_ε from polytopes of the form $C_g^{x,y}$ (by Definition 38 and Lemma 40), we can now construct an ITM P that has P_ε as its adversary polytope (up to transformation by a valid affine map). This ITM P is almost the protocol π we want to construct, except that π takes an valid affine map as an additional argument.

Lemma 48. *There exists a polynomial-time ITM P, such that for a set cover instance (n, m, S, d) and a valid affine map f, the following holds:*

- *The communication complexity of P upon input (n, m, S, d, f) is $O\big((\log n)^2 + \log m\big)$.*
- *The adversary-polytope of $P(n, m, S, d, f)$ is*

$$\mathbf{A}_{P(n,m,S,d,f)} = f(2^{-\lceil \log n \rceil} \cdot P_\varepsilon + \gamma)$$

where $\varepsilon := \frac{1}{nm+1}$ and $\gamma := m\varepsilon(1 - 2^{-\lceil \log n \rceil})u$ and $u := (1, \ldots, 1) \in \mathbb{R}^n$.

Proof. Let C be the ITM from Lemma 47.

Upon input of a set cover instance (n, m, S, d) and a valid affine map f, the ITM P behaves as follows:

- It chooses a random $j \in \{0, 1, \ldots, m\}$ with the following distribution: $j = 0$ has probability $1 - m\varepsilon$, and each $j \neq 0$ has probability ε.
- The value j is sent to the adversary.
- If $j = 0$, invoke $C(n, u, u, d, f)$.
- If $j > 0$, invoke $C(n, s_{i\cdot}, u, \|s_{i\cdot}\|_1 - 1, f')$ where f' is the affine map defined by $f'(v) := f(u - v)$.

Since $d \leq n = \|u \cdot u\|_1$, the ITM C is called with valid input in the case $j = 0$. Since $\|s_{i\cdot}\|_1 - 1 \leq \|s_{i\cdot} \cdot u\|_1$, and since f' is a valid affine map, the ITM C is always invoked with valid input in the case $j > 0$.

Since C is polynomial-time, so is P. The communication complexity of C is $O\big((\log n)^2\big)$, and sending the value j takes $O(\log m)$ bits, so the communication complexity of P is $O\big((\log n)^2 + \log m\big)$.

We now examine the adversary polytope of $P(n, m, S, d, f)$. We have

$$\begin{aligned}
\mathbf{A}_{P(n,m,S,d,f)} &\overset{(*)}{=} (1 - m\varepsilon)\mathbf{A}_{C(n,u,u,d,f)} + \sum_{i=1}^{m} \mathbf{A}_{C(n,s_{i\cdot},u,\|s_{i\cdot}\|_1-1),f'} \\
&\overset{(**)}{=} (1 - m\varepsilon) f(2^{-\lceil \log n \rceil} \cdot C_g^{u,u}) + \sum_{i=1}^{m} \varepsilon f'\big(2^{-\lceil \log n \rceil} \cdot C^{s_{i\cdot},u}_{\|s\cdot\|_1-1}\big) \\
&= (1 - m\varepsilon) f(2^{-\lceil \log n \rceil} \cdot C_g^{u,u}) + \sum_{i=1}^{m} \varepsilon f\big(u - 2^{-\lceil \log n \rceil} \cdot C^{s_{i\cdot},u}_{\|s\cdot\|_1-1}\big) \\
&= f\left((1 - m\varepsilon)2^{-\lceil \log n \rceil} \cdot C_g^{u,u} + \sum_{i=1}^{m} \varepsilon\big(u - 2^{-\lceil \log n \rceil} \cdot C^{s_{i\cdot},u}_{\|s\cdot\|_1-1}\big)\right) \\
&= f\left(2^{-\lceil \log n \rceil}\Big((1 - m\varepsilon)C_g^{u,u} + \sum_{i=1}^{m} \varepsilon\big(u - C^{s_{i\cdot},u}_{\|s\cdot\|_1-1}\big)\Big) + m\varepsilon(1 - 2^{-\lceil \log n \rceil})u\right) \\
&\overset{(***)}{=} f\left(2^{-\lceil \log n \rceil} \cdot P_\varepsilon + \gamma\right).
\end{aligned}$$

Here $(*)$ is an application of Lemma 44 and $(**)$ of Lemma 47. The equality $(***)$ follows from Lemma 40 and Definition 38. □

Using the tools given so far, we can finally give a construction of protocols π and ρ such that strongly good adversaries correspond to witness for a set cover instance. Thus finding a strongly good adversary is equivalent to solving a set cover instance. Note however that the protocol π constructed in the following theorem is not logarithmic, but $O((\log n)^2)$. We will solve this problem later by using shorter set cover instances.

Theorem 49. *There exist a function $\delta \in \Omega(\frac{1}{n^5 m})$ and polynomial-time ITMs π and ρ such that the following holds for every set cover instance (n, m, S, d):*

- *The communication complexity of $\pi(n, m, S, d)$ is in $O\big((\log n)^2 + \log m\big)$, that of $\rho(n, m, S, d)$ in $O(\log n)$. The ITM ρ is function-like.*
- *If (n, m, S, d) is a yes-instance, there is a strongly δ-good adversary for $\pi(n, m, S, d)$ and $\rho(n, m, S, d)$.*
- *If (n, m, S, d) is a no-instance, there is no strongly good adversary for $\pi(n, m, S, d)$ and $\rho(n, m, S, d)$.*

Furthermore, given black-box access to a strongly good adversary for $\pi(n, m, S, d)$ and $\rho(n, m, S, d)$, we can compute a witness for the set cover instance (n, m, S, d) in probabilistic polynomial time and with overwhelming probability.

Proof. For a vector $v \in \mathbb{R}^n$, let $f_n(v) := (\frac{1}{n}v, 1 - \sum_j \frac{1}{n}v_j) \in \mathbb{R}^{n+1}$. Let $u := (1, \dots, 1) \in \mathbb{R}^n$ and $\varepsilon_{n,m} := \frac{1}{nm+1}$ and $\xi_{n,m} := \frac{\varepsilon_{n,m}}{2(n^2+1)}$ and $\lambda_n := 2^{-\lceil \log n \rceil}$ and $\gamma_{n,m} := m\varepsilon_{n,m}(1 - \lambda_n)u$. Then for a vector $v \in \mathbb{R}^n$, let $w_{n,m}(v) := \frac{1}{2}(v + u - \gamma_{n,m})$.

Obviously, f_n is a valid affine map. To see that $f_n \circ w_{n,m}$ is a valid affine map, first note that $0 \le m\varepsilon_{n,m} \le 1$ and $0 \le \lambda_n \le 1$ and thus $0 \le \gamma_{n,m} \le u$. Then for $v \in [0,1]^n$, we have $\frac{1}{2}(v + u - \gamma_{n,m}) \le \frac{1}{2}(v + u) \le u$ and $\frac{1}{2}(v + u - \gamma_{n,m}) \ge \frac{1}{2}v \ge 0$. Thus $w_{n,m}([0,1]^n) \subseteq [0,1]^n$ and therefore $f_n \circ w_{n,m}$ is valid.

Let $X := \{e_1, \dots, e_n, -e_1, \dots, -e_n\}$ where the e_i are the unit vectors of $\mathbb{R}^n$. Then $\operatorname{conv} X$ is the cross-polytope that is the unit ball of $\|\cdot\|_1$. Let $B_n := (\frac{n}{2} - \xi_{n,m}) \cdot \operatorname{conv} X + \frac{1}{2}u$. Let

$$X_n := \tfrac{1}{2}\lambda_n\big((\tfrac{n}{2} - \xi_{n,m}) \cdot X + \tfrac{1}{2}u\big) + \tfrac{1}{2}u$$

Using $X \subseteq [-1,1]^n$, $\frac{n}{2} - \xi_{n,m} \in [0, \frac{n}{2}]$ and $\lambda_n \in [0, \frac{1}{n}]$, one verifies that $X_n \subseteq [0,1]^n$.

The ITMs π and ρ are constructed as follows. Upon input (n, m, S, d), the ITM π runs $P(n, m, S, d, f_n \circ w_{n,m})$. And for the same input, the ITM ρ runs $H(n, X_n, f_n)$.

Since (n, m, S, d) is a set cover instance, and $f_n \circ w_{n,m}$ is a valid affine map, P is called with valid input. And since $X_n \subseteq [0,1]^m$ and f_n is a valid affine map, the ITM H is also called with valid input.

Since P is polynomial time, so is π. And since H is polynomial-time and $\#X_n \in O(n)$, the ITM ρ is also polynomial-time. Since P has communication complexity $O\big((\log n)^2 + \log m\big)$, so does π. And since H has communication complexity $O(\log \#X_n) = O(\log n)$ and is function-like, so does ρ.

We now determine the adversary-polytope of $\pi(n, m, S, d)$. It is

$$\begin{aligned}\mathbf{A}_{\pi(n,m,S,d)} &= \mathbf{A}_{P(n,m,S,d,f_n\circ w_{n,m})} \\ &\overset{(*)}{=} f_n \circ w_{n,m}(\lambda_n \cdot P_{\varepsilon_{n,m}} + \gamma_{n,m}) \\ &= f_n(\tfrac{1}{2}\lambda_n \cdot P_{\varepsilon_{n,m}} + \tfrac{1}{2}u)\end{aligned}$$

where $(*)$ is shown by Lemma 48.

We now determine the adversary-polytope of $\rho(n, m, S, d)$. It is

$$\begin{aligned}\mathbf{A}_{\rho(n,m,S,d)} &= \mathbf{A}_{H(n,X_n,f_n)} \\ &\overset{(*)}{=} f_n(\operatorname{conv} X_n) \\ &= f_n(\tfrac{1}{2}\lambda_n \cdot B_n + \tfrac{1}{2}u)\end{aligned}$$

where $(*)$ follows from Lemma 46.

We will now show that if (n, m, S, d) is a yes-instance, there is a strongly δ-good adversary. Since $\varepsilon_{n,m} = \frac{1}{nm+1}$, by Lemma 39 the polytope $P_{\varepsilon_{n,m}}$ is an ε-set cover polytope. Thus, by Lemma 37, it is $\|P_{\varepsilon_{n,m}} - \frac{1}{2}u\|_1 = \frac{n}{2}$. Since $\|B_n - \frac{1}{2}u\|_1 = \frac{n}{2} - \xi_{n,m}$, there exists a vector $v \in P_{\varepsilon_{n,m}} - \frac{1}{2}u$ such that $d_1(v, B_n - \frac{1}{2}u) \geq \xi_{n,m}$. Then for $v' := \frac{1}{2}\lambda_n(v + \frac{1}{2}u) + \frac{1}{2}u$, we have $v' \in \frac{1}{2}\lambda_n P_{\varepsilon_{n,m}} + \frac{1}{2}u$ and $d_1(v', \frac{1}{2}\lambda_n B_n + \frac{1}{2}u) \geq \frac{1}{2}\lambda_n\xi_{n,m}$. For any two vectors v_a, v_b it is $d_1(f_n(v_a), f_n(v_b)) \geq d_1(\frac{1}{n}v_a, \frac{1}{n}v_b) = \frac{1}{n}d_1(v_a, v_b)$. So for $v'' := f_n(v')$ we have

$$v' \in \mathbf{A}_{\pi(n,m,S,d)} \qquad \text{and} \qquad d_1(v', \mathbf{A}_{\rho(n,m,S,d)}) \geq \frac{\lambda_n\xi_{n,m}}{2n}.$$

Since $v' \in \mathbf{A}_{\pi(n,m,S,d)}$, there exists an adversary A such that $\langle A, \pi(n, m, S, d)\rangle = v'$. Further, for any simulator S we have $\langle S, \rho(n, m, S, d)\rangle \in \mathbf{A}_{\rho(n,m,S,d)}$ and therefore the statistical distance between $\langle A, \pi(n, m, S, d)\rangle$ and $\langle S, \rho(n, m, S, d)\rangle$ is bounded from below by $\frac{1}{2}d_1(\langle A, \pi(n, m, S, d)\rangle, \langle S, \rho(n, m, S, d)\rangle) \geq \frac{\lambda_n\xi_{n,m}}{4n} = \delta(n, m)$. So A is a strongly $\delta(n, m)$-good adversary for $\pi(n, m, S, d)$ and $\rho(n, m, S, d)$. Since $\varepsilon_{n,m} \in \Omega(\frac{1}{nm})$ we have $\xi_{n,m} \in \Omega(\frac{1}{n^3m})$. Further, $\lambda_n \in \Omega(\frac{1}{n})$. So $\delta \in \Omega(\frac{1}{n^5m})$.

Now we prove that given black box access to a strongly good adversary A, we can efficiently compute a witness for (n, m, S, d). Let $v_A \in \mathbb{R}^{n+1}$ be the distribution of $\langle A, \pi(n, m, S, d)\rangle$. Since A is strongly good, it is $v_A \in \mathbf{A}_{\pi(n,m,S,d)} \setminus \mathbf{A}_{\rho(n,m,S,d)}$. Since v_A is a probability distribution, $f_n^{-1}(v_A)$ exists. Then

$$v'_A := \tfrac{2}{\lambda_n}(f_n^{-1}(v_A) - \tfrac{1}{2}u) - \tfrac{1}{2}u \in (P_{\varepsilon_{n,m}} - \tfrac{1}{2}u) \setminus (B_n - \tfrac{1}{2}u).$$

Since $B_n - \frac{1}{2}u = \{v \in \mathbb{R}^n : \|v\|_1 \leq (\frac{n}{2} - \xi_{n,m})\}$, we have $\|v'_A\|_1 > \frac{n}{2} - \xi_{n,m}$.

Given black-box access to A, we can efficiently sample $\langle A, \pi(n, m, S, d)\rangle$. Then by Lemma 22, we can estimate a probability distribution $\tilde{v}_A \in \mathbb{R}^{n+1}$ such that with probability at least $\frac{2}{3}$ we have $d_1(\tilde{v}_A, v_A) \leq \frac{\lambda_n}{2n}\xi_{n,m}$. (Note for this that $\frac{\lambda_n}{2n}\xi_{n,m}$ is noticeable.)[3]

[3] Strictly speaking, the formulation of Lemma 22 only guarantees that random variables of logarithmic length can be sampled if they are efficiently constructible, but does not cover the case when the random variables are efficiently constructible using an oracle (in this case A). However, it is easy to see that the proof of Lemma 22 relativises and therefore also applies to the present situation.

Since $\tilde{v}_A$ is a probability distribution, $f_n^{-1}(v_A)$ is defined. Then let $\tilde{v}_A := \frac{2}{\lambda_n}(f_n^{-1}(\tilde{v}_A) - \frac{1}{2}u) - \frac{1}{2}u$. For any two vectors v_a, v_b it is $d_1(f_n(v_a), f_n(v_b)) \geq d_1(\frac{1}{n}v_a, \frac{1}{n}v_b) = \frac{1}{n}d_1(v_a, v_b)$. Thus $d_1(f_n^{-1}(v_A), f_n^{-1}(\tilde{v}_A)) \leq nd_1(v_A, \tilde{v}_A) \leq \frac{\lambda_n}{2}\xi_{n,m}$. Therefore we have $d_1(v'_A, v_A) \leq \xi_{n,m}$. Since $\|v'_A\|_1 > \frac{n}{2} - \xi_{n,m}$ we have $\|\tilde{v}'_A\|_1 > \frac{n}{2} - 2\xi_{n,m} = \frac{n}{2} - \frac{\varepsilon_{n,m}}{n^2+1}$. Since further $d_1(\tilde{v}'_A, P_{\varepsilon_{n,m}}) \leq d_1(\tilde{v}'_A, v'_A) \leq \xi_{n,m} < \frac{\varepsilon_{n,m}}{n^2+1}$, and since $P_{\varepsilon_{n,m}}$ is an ε-set cover polytope, by Lemma 37 we can efficiently compute a witness for (n, m, S, d) from $\tilde{v}'_A$. So given black-box access to A we can compute a witness with probability at least $\frac{2}{3}$. Since set cover is in NP, we can efficiently verify the solution and therefore amplify the success probability by repetition.

It is left to show that if (n, m, S, d) is a no-instance, there is no strongly good adversary for π and ρ. Assume there was a strongly good adversary A. Then we can efficiently compute a witness using black-box access to A. In particular, such a witness exists. This is a contradiction to the assumption that (n, m, S, d) is a no-instance. □

Theorem 49 is formulated in terms of strongly good adversaries. However, to derive results about stand-alone security, we need to consider good adversaries. Fortunately, for the protocols π and ρ given in Theorem 49, these notions coincide as the following lemma shows.

Lemma 50. *Let π and ρ be ITMs. Assume that in an interaction with an ITM S, the ITM ρ sends its output to S at some point. Then A is a good adversary for (π, ρ) if and only if it is a strongly good adversary for (π, ρ).*

Moreover, the ITM ρ from Theorem 49 satisfies the above condition.

Proof. Since a strongly good adversary is always a good adversary, we only have to show that a good adversary for (π, ρ) is a strongly good adversary for (π, ρ).

Assume therefore that A is *not* a strongly good adversary. Then, by Definition 34 there exists an ITM S such that $\langle A, \pi\rangle$ and $\langle S, \rho\rangle$ have the same distribution. Define the random variables X and O by $(X, O) := \langle\langle A, \pi\rangle\rangle$, i.e., X denotes the output of the ITM A and O the output of the ITM π. For a string o, let μ_o be the distribution of X under the condition that $O = o$. We construct an ITM S' as follows: First, S' executes S. By assumption, in an interaction between S' and ρ, the ITM ρ sends the output it is going to give to S'. We can therefore assume that S' knows the output o of ρ when the interaction between the simulated S and ρ has finished. Then S' chooses a string a according to the distribution μ_o and outputs o. Since S' differs from S only in its output, the distributions of $\langle A, \pi\rangle$ and $\langle S', \rho\rangle$ are identical. And therefore by construction of μ_a it follows that $\langle\langle A, \pi\rangle\rangle$ and $\langle\langle S', \rho\rangle\rangle$, too. Therefore A is *not* a good adversary for (π, ρ). It follows that any good adversary for (π, ρ) is also strongly good.

The ITM ρ as constructed in the proof of Theorem 49 always sends its output to S (since ρ invokes the ITM H from Lemma 46 which sends its output by construction). So the conditions of this lemma are fulfilled for the ITM ρ from Theorem 49. □

Finally, we can deduce from Theorem 49 that finding good adversaries is hard (given a realistic complexity assumption).

Corollary 51. *If $\mathrm{NP} \not\subseteq \mathrm{BPTIME}(n^{O(\log n)})$, the following holds for all $\varepsilon > 0$:*

There is no efficient probabilistic algorithm that finds a good adversary for a pair of polynomial-time algorithms with logarithmic communication complexity, even when they are guaranteed to have a strongly k^ε-good adversary.

Proof. Assume that $\mathrm{NP} \not\subseteq \mathrm{BPTIME}(k^{O(\log k)})$ and that we can efficiently find good adversaries under the conditions specified in the corollary.

Since set cover is NP-hard and $\mathrm{NP} \not\subseteq \mathrm{BPTIME}(k^{O(\log k)})$ we also have that set cover is not in $\mathrm{BPTIME}(k^{O(\log k)})$. (To see this note that for any polynomial p it is $p(k)^{O(\log p(k))} \subseteq k^{O(\log k)}$.) Then for any function $f \in 2^{\Omega(\sqrt{\log k})}$, set cover with instances of size at most $f(k)$ is hard for probabilistic algorithms running in polynomial time in k. Let $\delta(n,m)$ be as in Theorem 49. W.l.o.g., we can assume δ to be monotonous. (Simply replace δ by $\delta'(n,m) := \min_{(n',m')\leq(n,m)} \delta(n',m') \in \Omega(\frac{1}{n^5 m})$.) Since $\delta(n,m) \in \Omega(\frac{1}{n^5 m})$, there is a function $f \in 2^{\Theta(\sqrt{\log k})}$ such that $\delta(f(k),f(k)) > k^{\varepsilon}$. Then by Theorem 49 we can convert any set cover instance (n,m,S,d) of size at most $f(k)$ (in particular, $n,m \leq f(k)$) into a pair of protocols π and ρ with the following properties: The protocols run in polynomial time and their communication complexity is bounded in $O((\log f(k))^2) = O(\log k)$. Furthermore, they are guaranteed to have a strongly $\delta(n,m)$-good adversary. And given a strongly good adversary (as a black-box) we can efficiently compute a witness for (n,m,S,d). Since $\delta(n,m) > k^{\varepsilon}$, by assumption we can find a good adversary in probabilistic polynomial time in k. By Lemma 50, this adversary is then also strongly good. So summarising, for set cover instances of length at most $f(k)$ we can find witnesses in probabilistic polynomial time in k. Since $f \in 2^{\Theta(\sqrt{\log k})}$ this is a contradiction to the fact that set cover with instances of size at most f is hard.

So our assumption is disproved. □

Although this hardness result is interesting in its own right, it does not yet show that there are computationally secure protocols that are not statistically secure. The first problem is that a protocol that has no good adversaries is not necessarily secure: it may be that there are adversaries that necessitate superpolynomial-time simulators. This problem will be solved by showing that (at least for the protocols π and ρ we constructed in the reduction) we can always efficiently compute a simulator. The second problem is that although it might be infeasible to compute a good adversary for a given protocol, it might still be possible that there *exists* an adversary that is good for all security parameters k. This we solve by using a stronger assumption which roughly states that there are efficiently computable sequences of NP-instances that are hard for polynomial-time machines. Then we can define a protocol that for each security parameter uses another such instance.

D.2 Separation of Computational and Statistical Security Without Auxiliary Input

In the preceding section we showed that finding a good adversary is hard. However, a good (and polynomial-time) adversary might still exist. To show that computational and statistical security fall apart (in the case without auxiliary input), we need an additional assumption:

Assumption 52. *There exists an sequence f_k of Boolean formulas computable in deterministic polynomial time that has the following two properties:*

- *Infinitely many f_k are satisfiable.*
- *For any probabilistic Turing machine A that runs in $n^{O(\log n)}$-time, the probability $\Pr[f_k(A(1^k)) = 1]$ is negligible in k.*

Although rarely written in this general form, Assumption 52 is a common assumption in cryptography. For example, a collision-resistant family $\{H_k\}_{k\in\mathbb{N}}$ of hash functions that is collision-resistant against uniform quasi-polynomial-time adversaries implies Assumption 52.

In the lemmas in this section, we will tacitly assume Assumption 52.

Given a sequence of hard instances as in Assumption 52, we can now construct protocols $\tilde{\pi}$ and $\tilde{\rho}$ that encode these hard instances as in Theorem 49. Note that the construction of Theorem 49 gives protocols with communication complexity $O((\log n)^2)$ where n is the length of the NP-instance, so we have to use sufficiently short instances to get protocols with logarithmic communication complexity.

Definition 53. *Let π and ρ be the ITMs from Theorem 49. Let f_k be as in Assumption 52. Let R be a witness-preserving reduction from SAT to set cover. Then let $\tilde{\pi}$ and $\tilde{\rho}$ be the following ITMs: Upon input 1^k, $\tilde{\pi}$ and $\tilde{\rho}$ run π and ρ, respectively, with input $R(f_{\lceil 2^{\sqrt{\log k}}\rceil})$.*

For the remainder of this section, $\tilde{\pi}$ and $\tilde{\rho}$ denote the ITMs from Definition 53.

The following lemma states that $\tilde{\pi}$ and $\tilde{\rho}$ are indeed suitable protocols for a separating example between computational and statistical stand-alone security in the case of logarithmic communication complexity.

Lemma 54. *The ITMs $\tilde{\pi}$ and $\tilde{\rho}$ run in polynomial-time and have logarithmic communication complexity. The ITM $\tilde{\rho}$ is function-like.*

Proof. By Theorem 49, the runtime of π and ρ with input $R(f_{\lceil 2^{\sqrt{\log k}}\rceil})$ is polynomial in the length of $R(f_{\lceil 2^{\sqrt{\log k}}\rceil})$. This again is polynomial in $\lceil 2^{\sqrt{\log k}}\rceil$ which is sublinear. So the runtime of $\tilde{\pi}$ and $\tilde{\rho}$ is polynomial in k.

Let $(n, m, S, d) := R(f_{\lceil 2^{\sqrt{\log k}}\rceil})$. By Theorem 49, the communication complexity of π and ρ with input $R(f_{\lceil 2^{\sqrt{\log k}}\rceil})$ is in $O\big((\log n)^2 + \log m\big) = O\big((\log\lceil 2^{\sqrt{\log k}}\rceil)^2\big) = O(\log k)$. Therefore the communication complexity of $\tilde{\pi}$ and $\tilde{\rho}$ is logarithmic in k.

Since ρ is function-like by Theorem 49, so is $\tilde{\rho}$. □

First we show that $\tilde{\pi}$ is not as secure as $\tilde{\rho}$ with respect to statistical stand-alone security.

Lemma 55. *The ITM $\tilde{\pi}$ is not as secure as the ITM $\tilde{\rho}$ with respect to statistical stand-alone security without auxiliary input.*

Proof. For a given $k \in \mathbb{N}$, let $\chi_k := 1$ if f_k is satisfiable, and $\chi_k := 0$ otherwise. Let $(n, m, S, d) := R(f_{\lceil 2^{\sqrt{\log k}}\rceil})$. Assume that $\chi_k = 1$. Then (n, m, S, d) is a yes-instance of set cover. Then, by Theorem 49, there is a strongly δ-good adversary A_k for $\pi(n, m, S, d)$ and $\rho(n, m, S, d)$ with $\delta \in \Omega(\frac{1}{n^5 m})$. Since n and m are polynomial in $\lceil 2^{\sqrt{\log k}}\rceil$, we have $\delta \in 2^{-O(\sqrt{\log k})} \subseteq \Omega(\frac{1}{k})$.

Let A be the ITM that upon input 1^k with $\chi_k = 1$ executes A_k. For $\chi_k = 0$ let A_k behave arbitrarily. Then $A(1^k)$ is a strongly $\chi_k\delta$-good adversary for $\tilde{\pi}(1^k)$ and $\tilde{\rho}(1^k)$. Since $\chi_k = 1$ holds infinitely often and $\delta \in \Omega(\frac{1}{k})$, we have that $\chi_k\delta$ is not negligible, so for every simulator S we have that $\langle A, \pi(1^k)\rangle$ and $\langle S, \rho(1^k)\rangle$ are statistically distinguishable. Then also $\langle\!\langle A, \pi(1^k)\rangle\!\rangle$ and $\langle\!\langle S, \rho(1^k)\rangle\!\rangle$ are statistically distinguishable.

Therefore π is not as secure as ρ with respect to statistical stand-alone security without auxiliary input. □

First we want to show that π is as secure as ρ with respect to computational stand-alone security. By Theorem 49 we will get that a polynomial-time adversary cannot be a strongly good adversary. This implies that there is a simulator for this adversary. However, we do not know that this simulator is also polynomial-time. The following lemma guarantees that this is at least the case for the protocols $\tilde{\pi}$ and $\tilde{\rho}$ constructed above.

Lemma 56. *There is a probabilistic polynomial-time oracle ITM S such that for every oracle ITM A and every $k \in \mathbb{N}$, the following holds: If A is not a strongly good adversary for $\tilde{\pi}(1^k)$ and $\tilde{\rho}(1^k)$, then $\langle\langle A, \tilde{\pi}(1^k)\rangle\rangle$ and $\langle\langle S^A(1^k), \tilde{\rho}(1^k)\rangle\rangle$ have the same distribution.*

Proof. Since S can fix the random tape of A, we can w.l.o.g. consider only deterministic ITMs A.

Since $\tilde{\pi}$ has logarithmic communication complexity, there are only a polynomial number of communication traces possible between A and $\tilde{\pi}(1^k)$ and a polynomial number of possible outputs of $\tilde{\pi}$. We call C the set of these traces and the set of outputs O. By construction of $\tilde{\pi}$, $O = \{1, \ldots, n\}$ for some n. For each $(c, o) \in C \times O$, the ITM S can efficiently compute the probability $P_{c,o}$ that the communication c and output o occur in an interaction between A and $\tilde{\pi}(k)$. (Note that this does not holds in general, since some of these probabilities might be efficiently constructible, but not efficiently computable. However, the reader may verify that for the protocol $\tilde{\pi}$ as constructed here, and using the fact that A is deterministic, it is indeed possible to efficiently compute $P_{c,o}$.) In particular, $P_{c,o} \in \mathbb{Q}$. Furthermore, for each communication $c \in C$, let a_c be the output of A after communication c.

Then $v_o := \sum_{c \in C} P_{c,o}$ is the probability that $\langle A, \tilde{\pi}(k)\rangle = o$. Then $v := (v_1, \ldots, v_n) \in \mathbb{Q}^n$. Since A is not strongly good, $v \in \mathbf{A}_{\tilde{\rho}(k)}$. We remember the construction of $\tilde{\rho}$ (see in particular Lemma 46): The ITM $\tilde{\rho}$ (or, more concretely, the ITM H simulated by $\tilde{\rho}$) expects a single message j of logarithmic length from S. Let J be the set of these possible messages. Then, for each $j \in J$ there is a probability distribution $v^{(j)}$ such that upon message j the ITM $\tilde{\rho}$ gives output i with probability $v_i^{(j)}$. Therefore the adversary-polytope of $\tilde{\rho}(k)$ has the form $\mathbf{A}_{\tilde{\rho}(k)} = \mathrm{conv}\{v^{(j)} : j \in J\}$. Note that S can efficiently compute all $v^{(j)}$.

So $v \in \mathrm{conv}\{v^{(j)} : j \in J\}$. Therefore there is a convex combination $v = \sum_{j \in J} r_j v^{(j)}$ with $\sum_j r_j = 1$ and $r_j \geq 0$. Since $\#J$ is polynomial in k, and v and $v^{(j)}$ are efficiently computable, we can efficiently compute the values r_j.[4]

Let S choose message j with probability r_j. Then $\langle S^A(1^k), \tilde{\rho}(1^k)\rangle = v$. So $\langle S^A(1^k), \tilde{\rho}(1^k)\rangle$ and $v = \langle A, \tilde{\pi}(1^k)\rangle$ have the same distribution.

To be able to investigate $\langle\langle S^A(1^k), \tilde{\rho}(1^k)\rangle\rangle$, we have to specify the output of S. Since a_c and $P_{c,o}$ for $c \in C$, $o \in O$ are known to S, it can efficiently compute the distribution of $\langle\langle A, \tilde{\pi}(1^k)\rangle\rangle$. (It is $\Pr[\langle\langle A, \tilde{\pi}(1^k)\rangle\rangle = (a, o)] = \sum_{c \in C} \delta(a_c = a) P_{c,o}$ where $\delta(a_c = a) = 1$ if and only if

[4] Finding the convex combination can be recast into a linear programming problem: Find a vector $(r_1, \ldots, r_{\#J})$ satisfying the linear equalities and inequalities $v_i = \sum_{j \in J} r_j v_i^{(j)}$ for all i, $\sum_j r_j = 1$, and $r_j \geq 0$ for all $j \in J$. Since these are polynomially many equations with rational coefficients, we can efficiently compute an exact solution using the ellipsoid method (see e.g., [GLS93, Theorem (6.4.9)]).

$a_c = a$ and 0 otherwise.) Since $\rho(1^k)$ sends its output to S (cf. in particular the construction of H from Lemma 46), we can assume that S knows the output o' of $\rho(1^k)$. Therefore S can compute the distribution of $(a, o) = \langle\!\langle A, \tilde{\pi}(1^k) \rangle\!\rangle$ under the condition $o = o'$. Then S chooses a according to that distribution and outputs a. Since $\langle S^A(1^k), \tilde{\rho}(1^k) \rangle$ and $\langle A, \tilde{\pi}(1^k) \rangle$ have the same distribution, it follows that the distributions of $\langle\!\langle S^A(1^k), \tilde{\rho}(1^k) \rangle\!\rangle$ and $\langle\!\langle A, \tilde{\pi}(1^k) \rangle\!\rangle$ are identical, too. □

Using this lemma we can now show that $\tilde{\pi}$ is indeed as secure as $\tilde{\rho}$ with respect to computational stand-alone security.

Lemma 57. *The ITM $\tilde{\pi}$ is as secure as the ITM $\tilde{\rho}$ with respect to computational stand-alone security without auxiliary input.*

Proof. Assume for contradiction that $\tilde{\pi}$ is not as secure as $\tilde{\rho}$ with respect to computational stand-alone security without auxiliary input. Then there exists a probabilistic polynomial-time ITM A such that for every polynomial-time ITM S the following distributions are computationally distinguishable

$$\langle\!\langle A(1^k), \tilde{\pi}(1^k) \rangle\!\rangle \quad \text{and} \quad \langle\!\langle S(1^k), \tilde{\rho}(1^k) \rangle\!\rangle. \tag{14}$$

Let K be the set of all $k \in \mathbb{N}$ such that $A(1^k)$ is strongly good for $\pi(1^k)$ and $\rho(1^k)$. By Lemma 56 there is a polynomial-time probabilistic oracle ITM S such that $\langle\!\langle A(1^k), \tilde{\pi}(1^k) \rangle\!\rangle$ and $\langle\!\langle S^{A(1^k)}(1^k), \tilde{\rho}(1^k) \rangle\!\rangle$ have the same distribution for all $k \notin K$. Since $A(1^k)$ runs in probabilistic polynomial time, $S^{A(1^k)}(1^k)$ does, too. So if K is finite, $\langle\!\langle A(1^k), \tilde{\pi}(1^k) \rangle\!\rangle$ and $\langle\!\langle S^{A(1^k)}(1^k), \tilde{\rho}(1^k) \rangle\!\rangle$ are computationally indistinguishable in contradiction to (14). Therefore K is infinite.

By Theorem 49, for all $k \in K$, given black-box access to $A(1^k)$ we can compute a witness w for $R(f_{\lceil 2^{\sqrt{\log k}} \rceil})$ in probabilistic polynomial time in k. Since R is a witness-preserving reduction, from w we can efficiently compute a witness (i.e., a satisfying assignment) for $f_{\lceil 2^{\sqrt{\log k}} \rceil}$. By outputting that witness, we can construct a probabilistic polynomial-time algorithm B with the property that for $k \in K$, the probability $\Pr[f_{\lceil 2^{\sqrt{\log k}} \rceil}(B(1^k)) = 1]$ is at least $\frac{2}{3}$.

For $n \in \mathbb{N}$, let $K_n := \{k \in \mathbb{N} : \lceil 2^{\sqrt{\log k}} \rceil = n\}$. Let C be the following Turing machine: Upon input 1^n, for each $k \in K_n$ it invokes $B(1^k)$ (it may invoke no instance of B if $K_n = \varnothing$). If one of these instances returns a witness for f_l, the machine C outputs that witness.

Since for $k > n^{\log n}$ it is $\lceil 2^{\sqrt{\log k}} \rceil > n$, all $k \in K_n$ satisfy $k \leq n^{\log n}$ and in particular, $\# K_n \leq n^{\log n}$. So $C(1^n)$ has running time in $\mathrm{poly}(n^{\log n}) = n^{O(\log n)}$.

For $k \in K$ and $n := \lceil 2^{\sqrt{\log k}} \rceil$, the Turing machine $C(1^n)$ calls $B(1^k)$ and thus gets a witness for $f_{\lceil 2^{\sqrt{\log k}} \rceil} = f_n$ with probability at least $\frac{2}{3}$. Since K is infinite, this happens for infinitely many n. So the probability $\Pr[f_n(C(1^n)) = 1]$ is at least $\frac{2}{3}$ for infinitely many n. This is a contradiction to Assumption 52. So our assumption that $\tilde{\pi}$ is not as secure as $\tilde{\rho}$ is wrong. □

Combining the results from this section, we get the separation of computational and statistical stand-alone security without auxiliary input.

Theorem 58 (Computational Does Not Imply Statistical Security Stand-Alone Security Without Auxiliary Input). *If Assumption 52 holds, computational stand-alone security without auxiliary input does not imply statistical stand-alone security without auxiliary input for*

polynomial-time protocols with logarithmic communication complexity, even if the ideal protocol is a function (i.e., a function-like ITM).

Proof. Immediate from Lemmas 54, 57 and 55. □

Because of Lemma 56, analogous theorems hold for black-box security and if we require the simulator to be efficient even in the statistical case.

One might ask whether this result also holds in the case with auxiliary input. Here we have to distinguish two cases, nonuniform and uniform auxiliary input. Definition 31 defines stand-alone security with nonuniform auxiliary input. In contrast, uniform auxiliary input in the sense of [Gol93] means that protocol-inputs and auxiliary input are chosen by a probabilistic polynomial-time machine. This allows to model additional information the adversary might have on the protocol-inputs and still enables sequential composition without introducing nonuniform complexity assumptions. If the protocols take no inputs (as is the case with $\tilde{\pi}$ and $\tilde{\rho}$), stand-alone security with uniform auxiliary input and stand-alone security without auxiliary input coincide since the adversary may choose the auxiliary input himself. Therefore Theorem 58 also applies in the case of stand-alone security with *uniform* auxiliary input. In the case of *nonuniform* auxiliary input however, our approach does not work, since we would need a variant of Assumption 52 that holds against nonuniform adversaries, which of course is impossible. In fact, in the next section we will show that Theorem 58 does not holds in the case with nonuniform auxiliary input.

D.3 The Stand-Alone Model With Auxiliary Input

We will now show that with nonuniform auxiliary input, computational security implies statistical security in the case of protocols with logarithmic communication complexity. This is done by showing that in this case any adversary's strategy can be encoded into an auxiliary input. The following lemma formalises this fact.

Lemma 59. *Let X and A be ITMs. Assume that X has communication complexity $O(\log k)$ upon input $(1^k, z)$. Then there is a polynomial-time ITM A_{poly} and a function f with $|f(k,z)| \in k^{O(1)}$ such that for all sequences x and z of strings, the distributions $\langle\!\langle A(1^k, z_k), X(1^k, x_k)\rangle\!\rangle$ and $\langle\!\langle A_{\mathrm{poly}}(1^k, f(k, z_k)), X(1^k, x_k)\rangle\!\rangle$ are statistically indistinguishable in k.*

Proof. For $k \in \mathbb{N}$, a string z and a sequence c of inputs and outputs of A, let $p^{msg}_{A,k,z}(c, m)$ denote the probability that $A(1^k, z)$ sends message m under the condition that its communication up to that point was c. Similarly, let $p^{out}_{A,k,z}(c, o)$ denote the probability that $A(1^k, z)$ terminates with output o under the condition that its communication up to that point was c. If these probabilities are undefined (because the communication c cannot occur with $A(1^k, z)$), we set $p^{msg}_{A,k,z}(c, m) = 0$ or $p^{out}_{A,k,z}(c, o) := 0$, respectively.

Let $A_{k,z}$ be the ITM that after communication c, sends m with probability $p^{msg}_{A,k,z}(c, m)$ and outputs o with probability $p^{out}_{A,k,z}(c, o)$. (If these probabilities do not add to 1, with the remaining probability $A_{k,z}$ terminates with a fixed output $\perp$.) Obviously, $\langle\!\langle A(1^k, z), X(1^k, x)\rangle\!\rangle$ and $\langle\!\langle A_{k,z}, X(1^k, x)\rangle\!\rangle$ have identical distributions for all strings x.

For $\alpha > 0$ and $r \in \mathbb{R}$, let $\lfloor r \rfloor_\alpha := \alpha \lfloor \frac{r}{\alpha} \rfloor$, i.e., $\lfloor r \rfloor_\alpha$ denotes r rounded down to a multiple of α.

Let $\tilde{p}^{msg}_{A,k,z}(c,m) := \lfloor p^{msg}_{A,k,z}(c,m) \rfloor_{2^{-k}}$ and $\tilde{p}^{out}_{A,k,z}(c,o) := \lfloor p^{out}_{A,k,z}(c,o) \rfloor_{2^{-k}}$.

Let $\tilde{A}_{k,z}$ be defined analogously to $A_{k,z}$, except that the probabilities $\tilde{p}^{msg}_{A,k,z}(c,m)$ and $\tilde{p}^{out}_{A,k,z}(c,o)$ are used. Then $\langle\!\langle A_{k,z}, X(1^k,x) \rangle\!\rangle$ and $\langle\!\langle \tilde{A}_{k,z}, X(1^k,x) \rangle\!\rangle$ are statistically indistinguishable in k. Since $A(1^k,z)$ has logarithmic communication complexity in k, and since the images of $\tilde{p}^{msg}_{A,k,z}(c,m)$ and $\tilde{p}^{out}_{A,k,z}(c,o)$ can be represented using k bit, there is a representation of $\tilde{p}^{msg}_{A,k,z}$ and $\tilde{p}^{out}_{A,k,z}$ whose length is polynomial in k. Let $f(k,z_k)$ be that representation. Given $f(k,z_k)$, we can simulate $\tilde{A}_{k,z}$ in polynomial time in k. Let then A_{poly} be the polynomial time ITM that upon input $(1^k, f(k,z))$ simulates $\tilde{A}_{k,z}$. Then for all sequences x and z of strings, $\langle\!\langle A(1^k,z_k), X(1^k,x_k) \rangle\!\rangle$ and $\langle\!\langle A_{\mathrm{poly}}(1^k, f(k,z_k)), X(1^k,x_k) \rangle\!\rangle$ are statistically indistinguishable in k. □

Since any adversary can be encoded into the auxiliary input of a polynomially-bounded one, it is not hard to show the following theorem which states that computational implies statistical stand-alone security with nonuniform auxiliary input in the case of logarithmic protocols.

Theorem 60 (Computational Implies Statistical Stand-Alone Security With Nonuniform Auxiliary Input). *Let π and ρ be polynomial-time ITMs. Assume that the communication complexity and the length of the output of π and ρ on input $(1^k, z)$ is logarithmic in k. If π is as secure as ρ with respect to computational stand-alone security with auxiliary input, then π is as secure as ρ with respect to statistical stand-alone security with auxiliary input.*

Proof. Assume that π is not as secure as ρ with respect to statistical security with auxiliary input. Then there exist an ITM A such that for every ITM S there are sequences x^S and z^S of strings polynomial length such that

$$\langle\!\langle A(1^k, z^S_k), \pi(1^k, x^S_k) \rangle\!\rangle \not\approx \langle\!\langle S(1^k, z^S_k), \rho(1^k, x^S_k) \rangle\!\rangle \tag{15}$$

where $\approx$ denotes statistical indistinguishability in k. Without loss of generality, we can assume that the communication complexity of A is logarithmic in k (since π has logarithmic communication complexity, too) and that the output of A is its view (and thus in particular has logarithmic length in k, too). Let $b(k)$ be an efficiently computable upper bound on the length of the output of A.

By Lemma 59 there exists a polynomial-time ITM A_{poly} and a function f with $|f(k,z)| \in k^{O(1)}$ such that for every ITM S, we have

$$\langle\!\langle A(1^k, z^S_k), X(1^k, x^S_k) \rangle\!\rangle \approx \langle\!\langle A_{\mathrm{poly}}(1^k, f(k, z^S_k)), X(1^k, x^S_k) \rangle\!\rangle.$$

From A_{poly}, we construct an ITM A^b_{poly} that upon input $(1^k, z)$ runs A_{poly} but truncates the output to length $b(k)$. Since $b(k)$ is an upper bound on the length of the output of A, we have for all ITMs S:

$$\langle\!\langle A(1^k, z^S_k), X(1^k, x^S_k) \rangle\!\rangle \approx \langle\!\langle A^b_{\mathrm{poly}}(1^k, f(k, z^S_k)), X(1^k, x^S_k) \rangle\!\rangle. \tag{16}$$

To finish the proof, we have to show that for any polynomial-time ITM S, there are sequences $\hat{z}^S$ and $\hat{x}^S$ of strings of polynomial length such that the families of distributions $real_S := \left\{\langle\!\langle A^b_{\mathrm{poly}}(1^k, \hat{z}^S_k), \pi(1^k, \hat{x}_k)\rangle\!\rangle\right\}_{k,\hat{z}^S_k,\hat{x}_k}$ and $ideal_S := \left\{\langle\!\langle S(1^k, \hat{z}^S_k), \rho(1^k, \hat{x}_k)\rangle\!\rangle\right\}_{k,\hat{z}^S_k,\hat{x}_k}$ are *not* computationally indistinguishable. Since A^b_{poly} has output of logarithmic length in k, we can assume w.l.o.g. that S has output of logarithmic length in k, too.

We define the (not necessarily polynomial-time) ITM S' which upon input $(1^k, z)$ invokes $S(1^k, f(k, z))$. Then by (16) and (15) and with $\hat{z}^S_k := f(k, z^{S'}_k)$ and $\hat{x}_k := x_k$, we have

$$\langle\!\langle A^b_{\mathrm{poly}}(1^k, \hat{z}^S_k), \pi(1^k, \hat{x}_k)\rangle\!\rangle \not\approx \langle\!\langle S'(1^k, z^{S'}_k), \rho(1^k, x_k)\rangle\!\rangle = \langle\!\langle S(1^k, \hat{z}^S_k), \rho(1^k, \hat{x}_k)\rangle\!\rangle.$$

In other words, the families of distributions $real_S$ and $ideal_S$ are *not* statistically indistinguishable. Since A^b_{poly}, S, π and ρ are polynomial-time ITMs, and have output of logarithmic length in k, by Theorem 23 the families $real$ and $ideal$ are *not* computationally indistinguishable, either. □

E Advantage-Based Security – Details and Proofs

Definition 61 (Advantage-Based Security). *Let B be an ITM and γ a function. We say that B is* γ-secure with respect to computational advantage-based security with auxiliary input *if for every polynomial-time ITM A and for all sequences x and z of strings of polynomial length, there is a negligible function μ such that* $\Pr[\langle A(1^k, z_k), B(1^k, x_k)\rangle = 1] \leq \gamma(k) + \mu(k)$ *for all* $k \in \mathbb{N}$.

We speak of statistical advantage-based security if the above holds with unbounded A.

We speak of advantage-based security without auxiliary input if A does not get the additional input z_k (i.e., the distribution $\langle A(1^k), B(1^k, x_k)\rangle$ is considered).

Definition 62 (Game of a Protocol). *Let B be an ITM. The* game $G^B_{k,n}$ of B, k, n *is the following one-player game:*

- *First, player 1 may choose a string x with $|x| \leq \log n$.*
- *Then, the game consists of the interaction $\langle A, B(1^k, x)\rangle$, where player 1 learns all messages that A receives, and may choose all message that A sends.*
- *The payoff of the game is* 1 *if B outputs* 1*, and* 0 *otherwise.*

If $B(1^k, x)$ has logarithmic communication complexity in $k + |x|$, the game-tree $G^B_{k,n}$ has polynomial size in $k + n$ (note that we do not claim that the representation of $G^B_{k,n}$ and in particular the probability distributions therein have polynomial size, since these distributions may contain irrational numbers).

Definition 63 (Distance of games). *Let G_1 and G_2 be two games in extensive form of the same structure (i.e., G_1 and G_2 differ only in their transition probabilities).*

For a path p, by $G_1(p)$ we denote the product of the probabilities associated with the chance-edges on the path p.

Then the distance $d(G_1, G_2)$ is defined as $\max_p |G_1(p) - G_2(p)|$ where p ranges over all paths in G_1 connecting the root and a leaf.

Lemma 64. *Let G_1 and G_2 be n-player games in extensive form of the same structure. Let $H_i(\mu_1, \dots, \mu_n)$ denote the expected payoff of player 1 for game G_i if the strategies $\mu_1, \dots, \mu_n$ are played.*[5]

Assume that H_i takes values in $[-1, 1]$. For mixed strategies $\mu_1, \dots, \mu_n$, it is $|H_1(\mu_1, \dots, \mu_n) - H_2(\mu_1, \dots, \mu_n)| \leq \#G_1 \cdot d(G_1, G_2)$.

Proof. For pure strategies $\sigma_1, \dots, \sigma_n$ we have $H_i(\sigma_1, \dots, \sigma_n) = \sum_{p \in S} G'(p) H(p)$. Here S is the set of all paths that are not ruled out by the strategies $\sigma_1, \dots, \sigma_n$, and $H(p)$ denotes the payoff $H(v)$ of the leaf v of path p. Since G_1 and G_2 have the same structure, $H(p)$ and S do not depend on i. Then

$$\begin{aligned}\big|H_1(\sigma_1, \dots, \sigma_n) - H_2(\sigma_1, \dots, \sigma_n)\big| &\leq \sum_{p \in S} \big|H(p)(G_1(p) - G_2(p))\big| \\ &\leq \#S \cdot d(G_1, G_2) \leq \#G_1 \cdot d(G_1, G_2).\end{aligned}$$

For mixed strategies $\mu_1, \dots, \mu_n$ we then have

$$\begin{aligned}\big|H_1(\mu_1, \dots, \mu_n) - H_2(\mu_1, \dots, \mu_n)\big| &= \big|\mathrm{E}_\mu[H_1(\sigma_1, \dots, \sigma_n) - H_2(\sigma_1, \dots, \sigma_n)]\big| \\ &\leq \mathrm{E}_\mu\big[|H_1(\sigma_1, \dots, \sigma_n) - H_2(\sigma_1, \dots, \sigma_n)|\big] \leq \#G \cdot d(G_1, G_2).\end{aligned}$$

Here $\mathrm{E}_\mu[X]$ denotes the expectation value of X if $\sigma_1, \dots, \sigma_n$ are chosen independently according to the distributions $\mu_1, \dots, \mu_n$. □

Lemma 65. *There is a deterministic polynomial-time ITM A^G such that the following holds: Let $n, k \in \mathbb{N}$. Let B be an ITM and let $G^B_{k,n}$ be the game of B, k, n. Let $\tilde{G}$ be a game with the same structure as $G^B_{k,n}$. Then*

$$\max_{|x| \leq \log n} \Pr[\langle A^G(\tilde{G}), B(1^k, x)\rangle = 1] \geq \max_{A,\, |x| \leq \log n} \Pr[\langle A, B(1^k, x)\rangle = 1] - 2\#G^B_{k,n} \cdot d(\tilde{G}, G^B_{k,n})$$

where the maxima go over strings x of length $|x| \leq \log n$ and over (possibly unbounded) ITMs A, and $G^B_{k,n}$ is given in extensive form.

Proof. In [KM92, Section 3.3] it is shown that for a one-player game G with perfect recall in extensive form, one can compute a pure strategy σ in deterministic polynomial time, such that σ is optimal in the following sense: For every mixed strategy μ we have $H(\sigma) \geq H(\mu)$. Here H denotes the payoff function of G.

Since $G^B_{k,n}$ has perfect recall by construction, the game $\tilde{G}$ has perfect recall, too. Then let A^G be the ITM that upon input $\tilde{G}$ computes an optimal pure strategy σ for $\tilde{G}$. The σ then prescribes the choice of a string x_σ with $|x_\sigma| \leq \log n$ and how to interact with B. Then A^G simply interacts with B as prescribed by the strategy σ (and ignores the choice of x_σ). Then, by definition of $G^B_{n,k}$ we have

$$\Pr\big[\langle A^G(\tilde{G}), B(1^k, x_\sigma)\rangle = 1\big] = H^B_{k,n}(\sigma)$$

[5] Although G_1 and G_2 have the same payoffs at corresponding leafs, their payoff functions may differ anyway since the payoff functions denote the expected payoff which change when the transition probabilities change.

where $H^B_{k,n}$ is the payoff function of $G^B_{k,n}$.

Let an ITM A and a string x_A with $|x|_A \leq \log n$ be given. Let μ be the mixed strategy that chooses $x := x_A$ as input for B and prescribes to send the messages A would send. Then

$$\Pr\big[\langle A, B(1^k, x_A)\rangle = 1\big] = H^B_{k,n}(\mu).$$

By using the fact that σ is an optimal strategy for $\tilde{G}$ and applying Lemma 64 twice, we get

$$\begin{aligned} H^B_{k,n}(\sigma) &\geq \tilde{H}(\sigma) - \#G^B_{k,z} \cdot d(\tilde{G}, G^B_{k,n}) \\ &\geq \tilde{H}(\mu) - \#G^B_{k,z} \cdot d(\tilde{G}, G^B_{k,n}) \\ &\geq H^B_{k,n}(\mu) - 2\#G^B_{k,z} \cdot d(\tilde{G}, G^B_{k,n}). \end{aligned} \tag{17}$$

where $\tilde{H}$ is the payoff function of $\tilde{G}$. So for all ITMs A and all strings x of length $|x_A| \leq \log n$, we have

$$\begin{aligned} \max_{|x|\leq \log n} \Pr\big[\langle A^G(\tilde{G}), B(1^k, x)\rangle = 1\big] &\geq \Pr\big[\langle A^G(\tilde{G}), B(1^k, x_\sigma)\rangle = 1\big] \\ &\overset{(17)}{\geq} \Pr\big[\langle A, B(1^k, x_A)\rangle = 1\big] - 2\#G^B_{k,z} \cdot d(\tilde{G}, G^B_{k,n}). \end{aligned}$$

From this, the lemma follows. □

Definition 66 (Efficiently playable games). *Let $\{G_{k,n}\}_{k,n\in\mathbb{N}}$ be a family of m-player games in extensive form. We call $\{G_{k,n}\}_{k,n\in\mathbb{N}}$ efficiently playable if the following two conditions hold:*

- *There is a deterministic polynomial time algorithm which upon input $(1^k, 1^n)$ computes the extensive form of $G_{k,n}$ excluding the probabilities at the chance nodes (i.e., we get the game-tree without transition probabilities, the information sets and the value of the payoff function on the leafs of the game tree).*
- *There is a probabilistic polynomial time algorithm R with the following property. Let $\sigma_1, \ldots, \sigma_m$ be pure strategies for players $1, \ldots, m$. For a path p from the root of $G_{k,n}$ to a leaf of $G_{k,n}$, let P be the probability that this path is played given the strategies $\sigma_1, \ldots, \sigma_m$. Then upon input $(1^k, 1^n, \sigma_1, \ldots, \sigma_m)$ the algorithm R outputs the path p with probability P.*

Lemma 67 (Estimating game trees). *Assume that $\{G_{k,n}\}$ is an efficiently playable family of games.*

Then there is a probabilistic polynomial-time algorithm T such that for every superpolynomial function f there is a negligible function δ such that the following holds: Upon input $(1^k, 1^n, 1^{f(k)})$, the algorithm T outputs a game tree $\tilde{G}$ that has the same structure as $G_{k,n}$, and with probability at least $1 - \delta(k)$, it is $d(\tilde{G}, G_{k,n}) \leq \delta(k)$.

Proof. Upon input $(1^k, 1^n, 1^f)$ for some $k, n, f \in \mathbb{N}$, the algorithm T proceeds as follows:

- Since $G_{k,n}$ is efficiently playable, we can compute the extensive form of $G_{k,n}$ with exception of the probabilities at the chance nodes. To get a complete extensive form, we have to estimate the probability distributions on the outgoing edges of each change node n.

– For each chance node v, let $(\sigma_1, \ldots, \sigma_m)$ be pure strategies not ruling out v. By $X_{k,n,v}$ we denote the following distribution: In a play of $G_{k,n}$ with pure strategies $(\sigma_1, \ldots, \sigma_m)$, if v is reached, let $X_{k,n,v}$ denote the node reached immediately after v, and if v is not reached, let $X_{k,n,v} := \bot$.
– Since $G_{k,n}$ is efficiently playable, there are at most polynomially-many nodes in $k + n$ (otherwise the extensive form of $G_{k,n}$ could not be computed in polynomial time). Therefore a node of $G_{k,n}$ can be represented using logarithmic length, so we can consider $X_{k,n,v}$ to be a random variable with logarithmic length in $k + n + |v|$. Then by Lemma 22, there is a probabilistic polynomial-time algorithm S_X such that upon input $(1^k, 1^n, v, 1^f)$ it outputs the description of a probability distribution $\tilde{X}$ such that with probability at least $1 - \frac{1}{f}$, we have $\Delta(X_z; \tilde{X}) \leq \frac{1}{f}$.
– For each chance node n, we therefore call $S_X(1^k, 1^n, v, 1^f)$ to get an estimate $\tilde{X}_{k,n,v}$ of $X_{k,n,v}$. Then we annotate each edge from v to a successor v' with the probability $\frac{\Pr[\tilde{X}_{k,n,v}=v']}{\Pr[\tilde{X}_{k,n,v}\neq\bot]}$. If $\Pr[\tilde{X}_{k,n,v} \neq \bot] = 0$, we assign an arbitrary probability distribution to the outgoing edges of v.
– Let $\tilde{G}$ be the resulting game and output $\tilde{G}$.

Since for each chance node v, the probabilities $\frac{\Pr[\tilde{X}_{z,n}=n']}{\Pr[\tilde{X}_{z,n}\neq\bot]}$ sum to 1, the output $\tilde{G}$ of algorithm T always is a game.

Let now f be a superpolynomial function in $|z|$. Assume that T is called with inputs $(1^k, 1^n, 1^{f(k)})$. We will show that the estimate $\tilde{G}$ output by $S_G(z, 1^{f(|z|)})$ has with overwhelming probability negligible distance from G_z. Let $k, n \in \mathbb{N}$ be fixed. By Lemma 22, for each chance node v, with probability $1 - \frac{1}{f(k)}$ we have $\Delta(X_{k,n,v}; \tilde{X}_{k,n,v}) \leq \frac{1}{f(k)}$. Therefore, with probability at least $1 - \frac{\#G_{k,n}}{f(k)}$ we have

$$\Delta(X_{k,n,v}; \tilde{X}_{k,n,v}) \leq \frac{1}{f(k)} \qquad \text{for all chance nodes } v \text{ in } G. \tag{18}$$

In the following, we assume that (18) holds.

Fix some path p in $G_{k,n}$ from root to leaf. We will show that $|G_{k,n}(p) - \tilde{G}(p)| \leq 4\#G_{k,n}\sqrt{1/f(k)}$, and since this holds for all paths p, it follows $d(G_{k,n}, \tilde{G}) \leq 4\#G_{k,n}\sqrt{1/f(k)}$.

Let l denote the length of the path p (i.e., the number of edges on the path). For any $i \leq l$ let $Q_i := \frac{\Pr[X_{k,n,p_i}=p_{i+1}]}{\Pr[X_{k,n,p_i}\neq\bot]}$. Here p_i is the i-th node on the path p. Similarly, let $\tilde{Q}_i := \frac{\Pr[\tilde{X}_{k,n,p_i}=p_{i+1}]}{\Pr[\tilde{X}_{k,n,p_i}\neq\bot]}$. By definition of $X_{k,n,v}$ we have $\Pr[X_{k,n,p_{i-1}} = p_i] = \Pr[X_{k,n,p_i} \neq \bot]$ and $\Pr[X_{k,n,p_0} \neq \bot] = 1$, so that $R_i := Q_1 \cdots Q_{i-1} = \Pr[X_{k,n,p_{i-1}} = p_i]$. In particular, $Q_1 \cdots Q_l = G(p)$. Further, $\tilde{Q}_1 \cdots \tilde{Q}_l = \tilde{G}(p)$ by construction of $\tilde{G}$.

We show that $R_i|Q_i - \tilde{Q}_i| \leq 4\sqrt{1/f(k)}$. If $R_i \leq \sqrt{1/f(k)}$, this follows directly from the fact that $Q_i, \tilde{Q}_i \in [0, 1]$. On the other hand, if $R_i \geq \sqrt{1/f(k)}$, by (18) there are $\mu, \nu \in \mathbb{R}$ with $|\mu|, |\nu| \leq 1/f(k)$ such that we have with $P_i := \Pr[X_{k,n,p_i} = p_{i+1}] \leq 1$:

$$R_i|Q_i - \tilde{Q}_i| \overset{(18)}{=} R_i \left|\frac{P_i}{R_i} - \frac{P_i + \mu}{R_i + \nu}\right| = \left|\frac{\nu P_i - \mu R_i}{R_i - \nu}\right| \leq \frac{f(k)^{-1} + f(k)^{-3/2}}{f(k)^{-1/2} - f(k)^{-1}} \overset{(*)}{\leq} \frac{2f(k)^{-1}}{f(k)^{-1/2}/2} \leq 4\sqrt{1/f(k)}.$$

In $(*)$ we used that w.l.o.g. we can assume $f(k) > 4$.

Since $\tilde{Q}_i \leq 1$ for all i, from $R_i|Q_i - \tilde{Q}_i| \leq 4\sqrt{1/f(k)}$ it follows that $\big|Q_1 \cdots Q_{i-1} Q_i \tilde{Q}_{i+1} \cdots \tilde{Q}_l - Q_1 \cdots Q_{i-1} \tilde{Q}_i \tilde{Q}_{i+1} \cdots \tilde{Q}_l\big| \leq 4\sqrt{1/f(k)}$ and therefore

$$\big|G_{k,n}(p) - \tilde{G}(p)\big| = \big|Q_1 \cdots Q_l - \tilde{Q}_1 \cdots \tilde{Q}_l\big| \leq 4l\sqrt{1/f(k)} \leq 4\#G_{k,n}\sqrt{1 - f(k)}.$$

Since this holds for all p, we have $d(G_{k,n}, \tilde{G}) \leq 4\#G_{k,n}\sqrt{1/f(k)}$ (under the assumption made above that (18) holds). Summarizing, for any k, n with probability at least $1 - \frac{\#G_{k,n}}{f(k)}$ we have $d(G_{k,n}, \tilde{G}) \leq 4\#G_{k,n}\sqrt{1/f(k)}$. By setting $\delta := 4\#G_{k,n}\sqrt{1/f(k)} \leq \frac{\#G_{k,n}}{f(k)}$ the lemma follows. □

Lemma 68. *Let B be a polynomial-time ITM with logarithmic communication complexity in the length of its input.*

There is a polynomial-time ITM A^T such that for every superpolynomial function f and every polynomial function n there is a negligible function μ such that

$$\max_{|x| \leq \log n(k)} \Pr\big[\langle A^T(1^k, 1^{n(k)}, 1^{f(k)}), B(1^k, x)\rangle = 1\big] \geq \max_{A,\, |x| \leq \log n(k)} \Pr\big[\langle A, B(1^k, x)\rangle = 1\big] - \mu(k).$$

Here the maxima go over strings x of length $|x| \leq \log n(k)$ and over (possibly unbounded) ITMs A.

Proof. Let $G^B_{k,n}$ be the game of B, k, n. Since there are only a polynomial number (in n) of strings x with length $|x| \leq \log n$, and since B has logarithmic communication complexity, the game tree of $G^B_{k,n}$ has polynomial size $\#G^B_{k,n}$ in $k + n$ and can be efficiently computed (with exception of the probabilities occurring in the game tree). Thus, since B runs in polynomial-time, the family $\{G^B_{k,n}\}_{k,n}$ of games can be efficiently played. Therefore, by Lemma 67 there is a probabilistic polynomial-time algorithm T such that for any superpolynomial function f there is a negligible function δ_f such that upon input $(1^k, 1^n, 1^{f(k)})$, the algorithm T outputs a grame tree $\tilde{G}$ such that with probability at least $1 - \delta_f(k)$ it is $d(\tilde{G}, G^B_{k,n}) \leq \delta_f(k)$.

Let A^G be the ITM from Lemma 65. Then define A^T to be the ITM that on input $(1^k, 1^n, 1^f)$ computes $\tilde{G} := T(1^k, 1^n, 1^f)$ and then executes $A^G(\tilde{G})$.

Let f be a superpolynomial function and n a polynomial function. Then by Lemma 65 we have

$$\begin{aligned}\max_{|x| \leq \log n} &\Pr\big[\langle A^T(1^k, 1^{n(k)}, 1^{n(k)}), B(1^k, x)\rangle = 1\big] \\ &\geq \max_{A,\, |x| \leq \log n} \Pr\big[\langle A, B(1^k, x)\rangle = 1\big] - 2\#G^B_{k,n(k)} \cdot \mathrm{E}[d(\tilde{G}_k, G^B_{k,n})]. \quad (19)\end{aligned}$$

Here $\tilde{G}_k$ denotes the game computed by $T(1^k, 1^{n(k)}, 1^{n(k)})$, and $\mathrm{E}[d(\tilde{G}, G^B_{k,n})]$ denotes the expected value of $d(\tilde{G}, G^B_{k,n})$.

Since with probability at least $1 - \delta_f(k)$ we have $d(\tilde{G}, G^B_{k,n(k)}) \leq \delta_f(k)$, it is $\mathrm{E}[d(\tilde{G}_k, G^B_{k,n(k)})] \leq 2\delta_f(k)$ negligible. Since $\#G^B_{k,n(k)}$ is polynomially bounded in $k + n(k)$, which is again polynomial in k, it follows that $\mu(k) := 2\#G^B_{k,n(k)} \cdot \mathrm{E}[d(\tilde{G}_k, G^B_{k,n})]$ is negligible. With (19) the lemma follows. □

Theorem 69. *Let B be a polynomial-time ITM that upon input $(1^k, x)$ has logarithmic communication complexity in k and reads only a prefix of x of logarithmic length in k. Let γ be a function.*

Assume that B is γ-secure with respect to computational advantage-based security without auxiliary input. Then B is γ-secure with respect to statistical advantage-based security without auxiliary input

The same holds for advantage-based security with auxiliary input.

Proof. We first examine the case of security without auxiliary input. We assume that B is *not* γ-secure with respect to statistical advantage-based security without auxiliary input.

Then there is an efficiently computable polynomially-bounded function n and a (possibly unbounded) ITM A and a sequences x of strings of length at most $\log n$ such that

$$\mathrm{Adv}(A, k, x_k) := \max\{0, \Pr[\langle A, B(1^k, x_k)\rangle = 1] - \gamma\}$$

is *not* negligible in k.

Let A^T be defined as in Lemma 68. For an integer $f \in \mathbb{N}$, let $\Delta_f(k) := \max_{|x|\leq \log n(k)} \mathrm{Adv}(A^T(1^k, 1^{n(k)}, 1^f), k, x)$. We extend this to functions f by setting $\Delta_f(k) := \Delta_{f(k)}(k)$. Then, for any superpolynomial function f, by Lemma 68 we have that $\Delta_f(k) \geq \mathrm{Adv}(A, k, x_k) - \mu(k)$ for some negligible function μ. Thus Δ_f is not negligible. Let P be the set of all positive polynomials with integer coefficients. Assume that for every polynomial $p \in P$, the function Δ_p is negligible. We say a function μ^* asymptotically dominates a function μ if for all sufficiently large k, we have $\mu^*(k) \geq \mu(k)$. In [Bel02] it is shown that for every countable set N of negligible functions there is a negligible function μ^* that asymptotically dominates every $\mu \in N$. Since P is countable, it follows that there is a negligible function μ^* that asymptotically dominates every Δ_p with $p \in P$.

Let $f(k) := \max\{f \in \mathbb{N} : \Delta_f(k) \leq \mu^*\}$. Then $\Delta_f \leq \mu^*$ and therefore negligible. Furthermore, we show that f is superpolynomial. For contradiction, assume that f was not superpolynomial. Then there exists a polynomial $p \in P$ such that $f(k) < p(k)$ for infinitely many k. By construction of f, we then have $\Delta_p(k) > \mu^*(k)$ for infinitely many k. This is a contradiction to the fact that μ^* asymptotically dominates Δ_p, so f is superpolynomial. But we have shown above that for every superpolynomial f, the function Δ_f is not negligible. So our assumption was wrong and there exists a polynomial p such that Δ_p is not negligible. In other words:

$$\max_{|x|\leq \log n(k)} \mathrm{Adv}(A^T(1^k, 1^{n(k)}, 1^{p(k)}), k, x)$$

is not negligible. For each k, let x_k be a string x for which the maximum is reached. Further, let $A^*(1^k)$ be the ITM that executes $A^T(1^k, 1^{n(k)}, 1^{p(k)})$. Then

$$\max\{0, \Pr[\langle A^*(1^k), B(1^k, x_k)\rangle = 1] - \gamma(k)\}$$

is not negligible, so there is no negligible function μ such that $\Pr[\langle A^*(1^k), B(1^k, x_k)\rangle = 1] \leq \gamma + \mu$. Since A^* runs in polynomial-time, this shows that B is *not* γ-secure with respect to computational advantage-based security without auxiliary input. This concludes the case of advantage-based security without auxiliary input.

The proof carries over to the case of advantage-based security with auxiliary input almost verbatim. The only change necessary is to supply the ITM A^* with an additional argument z which A^* ignores. □

References

[BCK98] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. A modular approach to the design and analysis of authentication and key exchange protocols. In *Proc. 30th Annual ACM Symposium on Theory of Computing (STOC)*, pages 419–428, 1998.

[Bea91] Donald Beaver. Secure multiparty protocols and zero knowledge proof systems tolerating a faulty minority. *Journal of Cryptology*, 4(2):75–122, 1991.

[Bel02] Mihir Bellare. A note on negligible functions. *Journal of Cryptology*, 15(4):271–284, 2002. Online available at `http://eprint.iacr.org/1997/004`.

[BPW04] Michael Backes, Birgit Pfitzmann, and Michael Waidner. Secure asynchronous reactive systems. IACR Cryptology ePrint Archive 2004/082, March 2004. To appear in *Information and Computation*.

[BU06] Michael Backes and Dominique Unruh. Three-dimensional interactive example of the construction of the set cover polytope. Online at `http://www.infsec.cs.uni-sb.de/~unruh/logarithmic-protocols/setcover.html`, 2006. Needs a browser with Java support.

[Can95] Ran Canetti. *Studies in Secure Multiparty Computation and Applications*. PhD thesis, Feinberg Graduate School of the Weizmann Institute of Science, Rehovot, June 1995. Online available at `http://www.wisdom.weizmann.ac.il/~oded/ran-phd.html`.

[Can00] Ran Canetti. Security and composition of multiparty cryptographic protocols. *Journal of Cryptology*, 3(1):143–202, 2000.

[Can01] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *42th Annual Symposium on Foundations of Computer Science, Proceedings of FOCS 2001*, pages 136–145. IEEE Computer Society, 2001. Full version online available at `http://www.eccc.uni-trier.de/eccc-reports/2001/TR01-016/revisn01.ps`.

[Can05] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. IACR ePrint Archive, December 2005. Full and revised version of [Can01], online available at `http://eprint.iacr.org/2000/067.ps`.

[CG99] Ran Canetti and Shafi Goldwasser. An efficient threshold public key cryptosystem secure against adaptive chosen ciphertext attack. In *Advances in Cryptology: EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 90–106. Springer, 1999.

[GL90] Shafi Goldwasser and Leonid Levin. Fair computation of general functions in presence of immoral majority. In *Advances in Cryptology: CRYPTO '90*, volume 537 of *Lecture Notes in Computer Science*, pages 77–93. Springer, 1990.

[GLS93] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, 2 edition, 1993.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28:270–299, 1984.

[GM95] Rosario Gennaro and Silvio Micali. Verifiable secret sharing as secure computation. In *Advances in Cryptology: EUROCRYPT '95*, volume 921 of *Lecture Notes in Computer Science*, pages 168–182. Springer, 1995.

[GMR88] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, 1988.

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game – or – a completeness theorem for protocols with honest majority. In *Proc. 19th Annual ACM Symposium on Theory of Computing (STOC)*, pages 218–229, 1987.

[Gol93] Oded Goldreich. A uniform-complexity treatment of encryption and zero-knowledge. *Journal of Cryptology*, 6(1):21–53, 1993. Extended version online available at `http://www.wisdom.weizmann.ac.il/~oded/PS/uniform.ps`.

[Gol98] Oded Goldreich. Secure multi-party computation. Department of Computer Science and Applied Mathematics, June 1998. Revised Version 1.4 October 2002, http://www.wisdom.weizmann.ac.il/users/oded/pp.htm.

[Gol04] Oded Goldreich. *Foundations of Cryptography – Volume 2 (Basic Applications)*. Cambridge University Press, May 2004. Previous version online available at http://www.wisdom.weizmann.ac.il/~oded/frag.html.

[HM00] M. Hirt and U. Maurer. Player simulation and general adversary structures in perfect multiparty computation. *Journal of Cryptology*, 13(1):31–60, 2000.

[KM92] Daphne Koller and Nimrod Megiddo. The complexity of two-person zero-sum games in extensive form. *Games and Economic Behavior*, 4(4):528–552, October 1992. Online available at http://theory.stanford.edu/~megiddo/pdf/recall.pdf (without figures).

[MR91] Silvio Micali and Phillip Rogaway. Secure computation. In *Advances in Cryptology: CRYPTO '91*, volume 576 of *Lecture Notes in Computer Science*, pages 392–404. Springer, 1991.

[Pap93] Christos M. Papadimitriou. *Computational Complexity*. Addison-Wesley, Reading, Menlo Park, New York, 1993.

[PW01] Birgit Pfitzmann and Michael Waidner. A model for asynchronous reactive systems and its application to secure message transmission. In *Proc. 22nd IEEE Symposium on Security & Privacy*, pages 184–200, 2001. Extended version of the model (with Michael Backes) IACR Cryptology ePrint Archive 2004/082, http://eprint.iacr.org/.

[Unr06] Dominique Unruh. Relations among statistical security notions or why exponential adversaries are unlimited, 2006. Online available at http://eprint.iacr.org/2005/406, submitted to Asiacrypt 2006.

[Yao82] Andrew C. Yao. Protocols for secure computations. In *Proc. 23rd IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 160–164, 1982.